

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Implémentation d'une base de données économiques régionale relationnelle

Devulder, Alain; Schmidt, R.

Award date:
1979

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

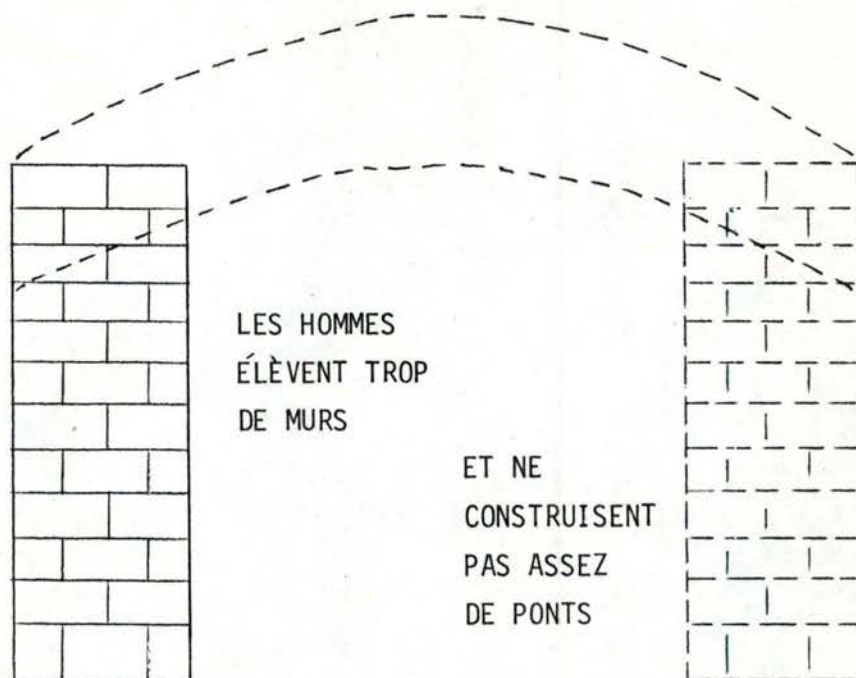
18/8/77

IMPLEMENTATION D' UNE BASE DE
DONNEES ECONOMIQUES REGIONALE
RELATIONNELLE

LBS 3212927



GS20-27014



NEWTON

REMERCIEMENTS

=====

Avant l'exposé de ce mémoire, nous tenons à remercier tous ceux qui, par leur amabilité et leur gentillesse, ont bien voulu nous faire profiter de leurs conseils et de leur expérience et qui, par là, ont contribué à la réalisation de ce travail.

Notre reconnaissance va surtout à J. Berleur s.j., à Mr. Moreau qui nous a accueilli dans le centre scientifique Poincaré d'IBM France à Paris et particulièrement au personnel de celui-ci : MM. J.P. Adam, O. Bertrand et J.J. Daudenarde.

Nous remercions également tout le personnel de l'Institut d'informatique et spécialement B. Lecharlier pour l'aide apportée lors de la rédaction du mémoire.

INTRODUCTION

Ce mémoire s'inscrit dans le cadre du Séminaire "L' Homme et l' informatique": Banques de données économiques dirigé par J. Berleur s.j.

Le but de cette recherche est la création d' une base de données économiques régionale afin de mettre à la disposition des pouvoirs publics et des entreprises manufacturières les informations leur permettant de prendre des décisions en matière de politique économique régionale. En cela, cette base de données peut être un outil de coordination et de concertation de l' initiative privée et du secteur public.

Toutefois, le volume des informations contenues dans cette base de données, le nombre élevé d' applications imaginables et le caractère mal défini de celles-ci, ont suggéré à (BDER) (1) l' idée d' organiser cette base de données comme un entrepôt de données très peu structuré où chaque utilisateur puisse prélever d' importants lots de données qu' il structurera en fonction de ses applications propres.

Cet entrepôt apparaît, ainsi, comme un 'outil-interface' entre les organismes collecteurs d' informations et ceux qui désirent exploiter celles-ci.

En fonction de cet objectif, nous résumerons, d' abord, la démarche de structuration des données, généralement admise dans la technique des banques de données. Elle consiste à définir quatre niveaux de description des données :

- le niveau infologique
- le niveau conceptuel
- le niveau d' implémentation logique
- le niveau d' implémentation physique.

Nous appliquerons, ensuite, cette démarche à la structuration de la base de données en nous inspirant de (BDER) et de (OCBD) pour la définition des trois premiers niveaux. En effet, d' une part nous avons dû apporter des modifications aux critères de structuration des données et d' autre part certains critères de structuration n' ont pas été appliqués à la base de données (BDER), avec toute la rigueur requise.

(1) J. BERLEUR *Une Banque de Données Economiques Régionale ? Aspects économique, informatique et socio-politique.*

Travaux de l' Institut d' Informatique n° 5 dans le cadre du Séminaire "L' homme et l' informatique" sous la Direction de J. Berleur s.j.

TABLE DES MATIERES

Introduction (A. Devulder et R. Schmit)	
1 Structure infologique du système (A. Devulder)	p. 1
1.1 Description du problème à résoudre	1
1.2 Modèle de représentation des informations	1
1.3 Exemples	3
2 Structure conceptuelle de la base de données (A. Devulder)	5
2.1 Définition du modèle relationnel	5
2.2 Passage du modèle 'entité-association' au modèle relationnel	5
2.3 Exemples	7
3 Implémentation logique de la base de données (A. Devulder)	9
3.1 Présentation des applications	9
3.2 Structuration pour l' application consultation	9
3.3 Structuration pour l' application mise à jour	10
3.3.1 Critère 'source de données'	10
3.3.2 Critère 'fréquence de collecte'	10
3.3.3 Critère 'donnée calculée'	11
3.3.4 Autres critères	11
3.3.5 Exemples	11
3.4 Accès logiques	14
3.5 Schéma de la base de données	15
3.6 Liste et brève description des opérations des 2 applications	16
3.6.1 Description de l' application consultation	17
3.6.1.1 Listage d' une relation logique	17
3.6.1.2 Projection d' une relation sur une liste de composantes	17
3.6.1.3 Sélection sur filtre	18
3.6.1.4 Jointure de deux relations	18
3.6.1.5 Adjonction de nouvelles composantes	19
3.6.2 Description de l' application mise à jour	20
3.6.2.1 Mise à jour massive de l' entrepôt des valeurs	20
3.6.2.2 Mise à jour ponctuelle du noyau	20
3.6.2.3 Mise à jour ponctuelle de l' entrepôt	20
3.6.2.4 Mise à jour ponctuelle du dictionnaire de données	21

4 Problèmes posés par la structuration de l' entrepôt (A. Devulder)	p. 22
4.1 Normalisation des relations	22
4.1.1 Définition des 3 formes normales et de certains mots clés	22
4.1.2 Intérêt de la 2ème forme normale	24
4.1.3 Mise sous 3ème forme normale des relations	25
4.1.3.1 Relation physique 'aide à l' investissement déjà attrib.'	25
4.1.3.2 Relation physique 'charge d' utilisation de potentiel hum.'	26
4.1.3.3 Relation physique 'volume des ventes'	27
4.1.3.4 Remarque	27
4.2 Précisions quant aux critères de structuration des données	28
4.3 Dimension de référence et fréquence de collecte	29
4.4 Modifications dans la structuration de l' entrepôt	29
4.4.1 Ajoute de deux relations à l' entrepôt	29
4.4.1.1 Relation 'indice du prix de détail'	30
4.4.1.2 Relation 'impôts indirects'	30
4.4.2 Modification d' une formule de calcul	31
4.4.3 Etablissement / entreprise	31
5 Implémentation physique de la base de données (A. Devulder)	32
5.1 Remarques préliminaires	32
5.1.1 La base de données descriptive (B.D.D.)	32
5.1.1.1 Description des relations de la B.D.D.	33
5.1.1.2 Justification des 5 modifications apportées à la B.D.D.	38
5.1.1.3 Utilité des relations de la base de données descriptive	39
5.1.2 Correspondance biunivoque entre une relation physique et un fich.	40
5.1.3 Incorporation de la notion de version aux relations	41
5.1.3.1 Données de type 'événement'	41
5.1.3.2 Données de type 'situation'	42
5.2 Création de la base de données	43
5.2.1 Principe général	43
5.2.2 Description d' un exemple de reformatage	43
5.3 Reconnaissance de l' utilisateur	46
5.4 Application de consultation	47
5.4.1 Consultation de la base de données	48
5.4.1.1 Listage d' une relation logique	49
5.4.1.2 Consultation sélective	53
5.4.1.2.1 Projection d' une relation	54
5.4.1.2.2 Sélection sur filtre	56

5.4.1.2.3 Jointure de deux relations	p. 58
5.4.1.2.4 Adjonction de nouvelles composantes	61
5.4.1.2.5 Autres opérations de consultation	61
5.4.1.2.6 Edition	62
5.4.2 Consultation de la base de données descriptive	64
5.4.2.1 Informations décrivant les relations de la B.D.D.	64
5.4.2.2 Listage d' une relation de la B.D.D.	64
5.4.2.3 Projection d' une relation de la B.D.D.	64
5.4.2.4 Sélection sur filtre	65
5.4.2.5 Jointure de deux relations	65
5.5 Mise à jour de la base de données	67
5.5.1 Mise à jour ponctuelle	67
5.5.1.1 Mise à jour ponctuelle de la base de données	68
5.5.1.1.1 Format général	68
5.5.1.1.2 Algorithme pour la mise à jour ponctuelle de la B.D.	68
5.5.1.1.3 Algorithme pour la mise à jour ponctuelle du noyau	69
5.5.1.1.4 Algorithme pour la mise à jour ponctuelle de l' entr.	70
5.5.1.2 Archivage des données de l' entrepôt	75
5.5.1.2.1 Archivage des données de type 'événement'	75
5.5.1.2.2 Archivage des données de type 'situation'	76
5.5.2 Contrainte de la présence obligatoire	77
5.5.3 Versions partielles et complètes	78
5.5.4 Données calculées	78
5.5.5 Mise à jour de la B.D.D. ou extensibilité du modèle	78
5.5.5.1 Ajoute d' une relation logique	79
5.5.5.2 Suppression d' une relation logique	81
5.5.5.3 Ajoute d' une relation physique	82
5.5.5.4 Suppression d' une relation physique	83
5.5.5.5 Ajoute d' une composante à une relation physique	85
5.5.5.6 Suppression d' une composante d' une relation physique	88
5.5.6 Mise à jour massive de l' entrepôt des valeurs	90
5.5.6.1 Algorithme de mise à jour massive de données 'événement'	91
5.5.6.2 Algorithme de mise à jour massive de données 'situation'	92
5.6 Problèmes particuliers de la gestion automatique de la B.D.	94
5.6.1 Les séries chronologiques	94
5.6.2 Les versions partielles	94
5.6.3 Données calculées	94
5.6.4 Le législateur et la base de données.	95
5.6.5 Autres problèmes	95

6 Implémentation réalisée lors du stage (R. Schmit)	p. 96
6.1 Description de l' outil utilisé	96
6.1.1 Une machine virtuelle	96
6.1.2 Un système d' exploitation	96
6.1.3 Un langage de programmation	97
6.1.4 Un langage d' interrogation de bases de données	97
6.2 Choix préliminaires	97
6.2.1 Subset de données	97
6.2.2 Le noyau	98
6.2.3 La dimension de référence	99
6.3 Présentation de notre réalisation	99
6.3.1 Les descriptif existant	100
6.3.1.1 Descriptif des relations	100
6.3.1.2 Descriptif des composantes	101
6.3.2 La base de données descriptive	101
6.4 Création physique de la base de données	103
6.4.1 Création des descriptifs des relations, composantes	103
6.4.2 Chargement des valeurs	104
6.4.3 Opérations à effectuer en vue de rendre la B.D. interrogeable	106
6.5 Application de consultation réalisée grâce au 'LAPIN'	107
6.5.1 Présentation du langage LAPIN	107
6.5.2 Listage d' une relation grâce au LAPIN	107
6.5.3 Projection d' une relation grâce au LAPIN	109
6.5.4 Sélection sur filtre grâce au LAPIN	111
6.5.5 Jointure de deux relations par LAPIN	113
6.5.6 Fonctions horizontales et verticales de LAPIN	114
6.5.7 Temps de réponse de LAPIN	115
6.5.8 Précautions pour réaliser des jointures en LAPIN	116
6.5.9 Combinaison des opérations de consultation grâce au LAPIN	117
6.5.10 Suggestions en vue d' améliorer le 'LAPIN'	117
6.5.10.1 Tris explicites ou implicites	118
6.5.10.2 Fonctions horizontales	118
6.5.10.3 Fonctions verticales	119
6.5.10.4 Relations intermédiaires	119
6.5.10.5 Jointure sur des composantes non adjacentes	119
6.5.10.6 Elimination des duplications	119
6.5.10.7 Contraintes d' intégrité	

6.6 Application de mise à jour	p. 120
6.6.1 Reconnaissance de l' utilisateur	120
6.6.2 Mise à jour ponctuelle de la base de données	121
6.6.2.1 Informations à fournir par le gestionnaire	121
6.6.2.2 Contrôles effectués lors de la mise à jour	122
6.6.2.3 Types de modifications	122
6.6.2.4 Exemple de procédure de m-à-j du noyau (modification)	123
6.6.2.5 Exemple de procédure de m-à-j du noyau (création)	125
6.6.2.6 Exemple de procédure de m-à-j de l' entrepôt (création)	126
6.6.2.7 Exemple de procédure de m-à-j de l' entrepôt (calcul)	128
7 Ebauche d' implémentation de la B.D. au moyen du Système R	132
7.1 Présentation du Système R	132
7.2 Points intéressants du Système R	133
7.2.1 Dictionnaire de données	133
7.2.2 Utilisation du langage SEQUEL	134
7.2.2.1 Facilités d' interrogation	134
7.2.2.1.1 La clause SELECT	134
7.2.2.1.2 La clause WHERE	134
7.2.2.1.3 La clause IN	135
7.2.2.1.4 La clause GROUP BY	136
7.2.2.1.5 Fonctions verticales	136
7.2.2.1.6 Intersection, union, différence	137
7.2.2.2 Facilités de manipulation	137
7.2.2.3 Facilités de définition	138
7.2.2.3.1 Création d' une nouvelle relation	138
7.2.2.3.2 Spécification d' aides d' accès : LINK	138
7.2.2.3.3 Définition de 'vues' de relations	138
7.2.2.4 Facilités de contrôle	139
7.2.2.4.1 Permission d' accès	139
7.2.2.4.2 Vérification automatique de prédicats	139
7.2.2.4.3 Déclenchement automatique de l' exécution d' instructions	140
7.3 Création de la base de données	140
7.4 Application de consultation	141
7.4.1 Listage d' une relation logique	142
7.4.2 Consultation sélective	144
7.4.2.1 Projection d' une relation	144
7.4.2.2 Sélection sur filtre	146

7.4.2.3 Jointure de deux relations	p. 147
7.4.2.4 Adjonction de nouvelles composantes	149
7.4.2.5 Autres opérations	150
7.4.2.6 Commandes composées	150
7.5 Application mise à jour	151
7.5.1 Mise à jour ponctuelle de la B.D.	151
7.5.1.1 'Programme' de mise à jour ponctuelle du noyau	152
7.5.1.2 'Programme' de mise à jour ponctuelle de l' entrepôt	153
7.5.1.2.1 'Programme' de mise à jour ponc. de données 'événement'	153
7.5.1.2.2 'Programme' de mise à jour ponc. de données 'situation'	154
7.5.2 Mise à jour ponctuelle de la B.D.D. - extensibilité	155
7.5.2.1 L' ajoute ou la suppression d' une relation	155
7.5.2.2 L' ajoute d' une composante à une relation physique	155
7.5.3 Mise à jour massive de l' entrepôt	156
7.5.3.1 'Programme' de mise à jour massive de données 'événement'	157
7.5.3.2 'Programme' de mise à jour massive de données 'situation'	157
7.6 Conclusions de l' étude du Système R	158
7.6.1 Facilités de description des données	158
7.6.2 Clause VIEW et TRIGGER	158
7.6.3 Système général de bases de données	159
Conclusion générale (A. Devulder et R. Schmit)	

Références

Bibliographie sommaire

Annexe 1 : ensemble des relations de la base de données

Annexe 2 : ensemble des descriptifs des relations de la B.D.

Annexe 3 : contenu des relations de la base de données descriptive

Annexe 4 : exemples de mise à jour de relations de la B.D.

On distingue généralement plusieurs niveaux de description et de structuration des informations contenues dans une base de données et ce, quels que soient les objectifs à réaliser au moyen de cette base de données. Nous allons résumer ci-dessous le rôle de ces différents niveaux, et, dans les chapitres suivants, nous appliquerons cette démarche à la base de données économiques régionale.

1) Niveau du réel perçu ou niveau infologique

La première chose à faire pour définir une base de données est de décrire avec la plus grande précision possible la partie du monde réel qu'elle devra représenter. Les situations qu'on est susceptible de représenter par une base de données sont trop variées et différentes les unes des autres pour qu'il soit facile de définir un formalisme général d'expression d'une situation réelle. Cependant, certains ont tenté de définir de tels formalismes. Le modèle 'entité-association' défini dans (CBTL) en est un exemple et est préconisé par (BDER) pour décrire la partie du monde réel à laquelle on est susceptible de s'intéresser pour prendre des décisions en matière de politique économique régionale. En général, on se satisfera, à ce niveau, d'une description en langage usuel.

2) Niveau conceptuel

De la description de la situation réelle à laquelle on s'intéresse, on extrait l'ensemble des informations à stocker dans la base de données qu'on exprimera dans un formalisme indépendant des applications à réaliser. On obtiendra, ainsi, la vue la plus stable possible des données. En effet, cette vue ne changera que si la situation réelle change mais pas si les applications sont modifiées. La description des données au niveau conceptuel constituera, pour cette raison, l'élément principal du système de documentation de la base de données (dictionnaire de données).

3) Niveau des accès logiques

A ce niveau on prend en compte l'ensemble des applications agissant sur la base de données. Ces applications seront réalisées par des programmes effectuant principalement des accès aux données. Pour une application déterminée, il est généralement possible de définir de nombreuses manières d'accéder aux données se distinguant par des coûts différents (par exemple : en volume occupé et en temps d'exécution).

La structuration au niveau des accès logiques consiste à définir les modes d'accès aux données de façon à minimiser globalement le coût des applications, sans toutefois entrer dans le détail de leur réalisation. (C'est pourquoi on parle d'accès logiques).

4) Niveau de l'implémentation physique

A ce niveau on réalise, à l'aide d'un software bien déterminé, l'implémentation des structures de données définies au niveau des accès logiques. Nous ne nous y attarderons pas à ce stade car la plus grande partie de ce travail y est consacrée. Disons cependant que nous présentons :

- les algorithmes généraux des applications définies sur la base de données
- les programmes réalisés au cours de notre stage au Centre Scientifique Poincaré d'IBM France à Paris
- la description, sur base de documents, de l'implémentation de la base de données à l'aide d'un prototype de gestion de bases de données relationnelles existant : le Système R.

REMARQUES

=====

- a) Il n'existe pas à l'heure actuelle de système général de base de données possédant des langages pour décrire les données à chacun des niveaux proposés. La démarche ci-dessus est donc avant tout méthodologique.
- b) Si on utilise un système existant on aura généralement des outils pour décrire les données au niveau des accès logiques et de l'implémentation physique et les programmes seront écrits d'après la description des données au niveau des accès logiques. Par contre, si on utilise des systèmes de fichiers plus rudimentaires, on devra souvent écrire les programmes au dernier niveau.
- c) Enfin, il est illusoire d'entièrement se tabler sur le caractère systématique de la démarche ci-dessus. En réalité, dans chaque cas réel, nombreux seront les choix pragmatiques ou arbitraires et les retours en arrière. Par exemple, la description conceptuelle des données sera toujours partiellement arbitraire et guidée par une certaine idée de ce que seront les applications à définir. De même, la notion d'accès logiques est assez vague et la minimisation globale du coût des applications une opération plus facile à énoncer qu'à effectuer.

CHAPITRE 1 STRUCTURE INFOLOGIQUE DU SYSTEME

Le niveau infologique s'attache à décrire la partie du monde réel à laquelle on s'intéresse. Le modèle de représentation des données utilisé est le modèle 'entité-association' défini dans (BBBC) et (CBTL). Il permet d'obtenir un schéma de description unique d'un réel perçu.

1.1 Description du problème à résoudre

Le réel perçu sera l'ensemble des activités économiques régionales des pouvoirs publics et le fonctionnement des entreprises manufacturières à moyen et à long terme.

A chacune des fonctions de politique économique, (BDER) a associé des activités qui en découlent. Afin de définir chacune de ces fonctions, des enquêtes ont été menées auprès de certaines entreprises pour connaître les informations qu'elles ont à fournir aux pouvoirs publics. De plus, (BDER) y a ajouté d'autres informations jugées pertinentes par des utilisateurs chargés d'élaborer la politique économique. Le processus d'élaboration a été identique pour "l'inventaire-entreprise". Méthodologiquement on est passé de la détermination des fonctions à la fixation des objectifs et ensuite à la définition des informations nécessaires à l'appréciation de ces objectifs.

1.2 Modèle de représentation des informations

Le modèle de représentation des informations préconisé par (BDER) est le modèle 'entité-association' défini dans (BBBC) et (CBTL) qui propose de représenter la structure des informations à partir de trois concepts : objet (entité), association et propriété.

Un OBJET du système ou ENTITE possède une existence en soi, est perçu comme un tout par un utilisateur et est caractérisé par des propriétés. Ainsi par exemple, *une entreprise, un produit*.

Un 'type d'entité' sera un ensemble d'entités de 'même nature', caractérisé par les mêmes propriétés et contenant toutes les entités existantes de cette nature. Par exemple, le type d'entité *entreprise* contiendra toutes les entreprises existant dans le cadre spatio-temporel qu'on s'est fixé.

Une ASSOCIATION est un ensemble de deux ou plusieurs objets où chacun d' eux joue un rôle déterminé. Ainsi, par exemple, entre les objets *entreprise* et *produit* peut exister une association *fabrication*.

Un 'type d' association' sera, alors, un ensemble d' associations de 'même nature', cet ensemble contenant, de plus, toutes les associations de cette nature. Par exemple, le type d' association *fabrication* contiendra toutes les associations formées d' une entreprise et d' un produit qu' elle fabrique.

Une PROPRIETE caractérisant une entité ou une association est une qualité que les individus attribuent à cette entité ou cette association.

Un 'type de propriété' est l' ensemble de toutes les valeurs de 'même nature' susceptibles de caractériser des entités ou des associations d' un type déterminé. Par exemple, *nom*, *adresse*, *numero de TVA* sont des types de propriétés de l' entité *entreprise*.

Certaines propriétés peuvent ne pas être indépendantes. On dira qu' elles sont liées par une ou plusieurs CONTRAINTE(S) D' INTEGRITE.

Enfin , par la suite, nous utiliserons souvent l' abus de langage consistant à parler d' entité, d' association ou de propriété au lieu de type d' entité, de type d' association ou de type de propriété particulièrement lorsque ce que l' on peut dire d' une entité, d' une association ou d' une propriété particulière s' applique aussi à toutes les autres du même type. Parmi les propriétés d' une entité ou d' une association, certaines ne varient pas au cours du temps, d' autres varient de manière plus ou moins continue. Etant donné l' objectif assigné à la base de données économiques régionale, il apparaissait nécessaire de tenir compte de ces variations, c' est à dire de définir non pas une base de données représentant la réalité économique régionale à une époque déterminée, mais bien une base de données historiques permettant d' étudier l' évolution de cette réalité.

Il est postulé dans (BDER) que l' on peut associer à chaque type de propriété une caractéristique dite de 'mouvement' indiquant la manière dont varient au cours du temps les propriétés de ce type. Définissons-la comme suit: une propriété caractérisée '*EVENEMENT*' est stable, c' est à dire qu' elle ne varie pas. Une entité ne possédant que des propriétés caractérisées '*EVENEMENT*' est souvent un objet apparu dans le système réel à un moment précis et dont toutes les propriétés sont entièrement et définitivement fixées à ce

moment. Ce qui justifie la terminologie 'EVENEMENT'.

Une propriété caractérisée 'SITUATION' varie de manière continue, c'est à dire qu' on peut la considérer comme stable durant une période de temps déterminée. La plupart des propriétés caractérisant des associations seront du genre 'SITUATION' car les associations entre entités varient elles-même souvent au cours du temps. Par exemple, une entreprise fabriquant un produit en produira des quantités différentes selon les époques et peut finalement en arrêter la fabrication auquel cas l' association disparaîtra.

Enfin, les propriétés caractérisées 'EXISTANT' ne varient que de manière occasionnelle et aléatoire, cfr (BDER p. 67).

Par exemple, un *numéro de brevet* aura la caractéristique 'EVENEMENT', le *nombre de travailleurs d' une entreprise* aura la caractéristique 'SITUATION' et *ses données signalétiques* auront la caractéristique 'EXISTANT'.

Lorsqu' une propriété possède la caractéristique de mouvement 'SITUATION', il apparaît nécessaire de définir avec exactitude la période de temps pour laquelle on peut se satisfaire de la connaissance d' une seule valeur (moyenne par exemple) de la propriété. Cette période de temps sera appelée la DIMENSION DE REFERENCE de la propriété (ou plus exactement, du type de propriété). On pourrait encore se poser la question de savoir si la notion de 'dimension de référence' a un sens pour les propriétés caractérisées 'EVENEMENT' ou 'EXISTANT'. Pour les premières, cette dimension de référence est infinie, puisqu' elles sont stables durant une période infinie. Quant aux secondes, il est impossible de leur associer une dimension de référence, puisqu' elles peuvent aussi bien ne jamais varier, qu' être complètement modifiées durant n' importe quelle période de temps.

1.3 Exemples

Prenons trois exemples d' entité dont nous suivrons l' évolution dans la structuration de la base de données.

EX 1 : Entité *caractéristiques générales de l' entreprise* ayant comme identifiant unique *identifiant de l' entreprise (id. EP)* et caractérisée par les propriétés suivantes :

110005 : Identifiant des établissements hors région
 110006 : Adresse de EP (siège social)
 110007 : Nom ou raison sociale de l' EP
 110008 : Date de création de l' EP (publication au Moniteur)
 110009 : Forme juridique de l' EP
 12000N : Secteur d' activité n de EP
 ces propriétés ont pour caractéristique de mouvement 'existant'

- 440001 : Nombre de mois de retard de paiements vis-à-vis de l' ONSS
- 440002 : Montant des paiements en retard vis-à-vis de l' ONSS
- 440003 : Nombre de mois de retard de paiement au Ministère des Finances, pour précompte professionnel.
- 440004 : Montant des paiements en retard vis-à-vis du Min. Fin. pour précompte professionnel
- 440005 : Nombre de mois de retard de paiements au Min. Fin. pour impôt des sociétés
- 440006 : Montant des paiements en retard vis-à-vis du Min. Fin. pour impôt des sociétés

ces dernières propriétés ont comme caractéristique de mouvement 'situation' et comme dimension de référence 'trimestre'.

EX 2 : Entité *capital passif étranger (emprunt obligataire)* ayant comme identifiant unique *identifiant de l' entreprise (id. EP)* et *identifiant emprunt obligataire (id. Emp. Obl.)* et caractérisée par les propriétés suivantes :

- 422032 : Montant
- 422033 : Taux d' intérêt
- 422034 : Terme
- 422035 : Date d' émission
- 422036 : Nombre d' obligations

ces propriétés ont la caractéristique mouvement 'événement'.

EX 3 : Entité *capital investi* ayant comme identifiant unique *identifiant de l' établissement (id. ET)* et *identifiant du type d' investissement (id. Typ. Inv.)* et caractérisée par les propriétés suivantes :

- 324002 : Valeur de l' augmentation de capital
- 324003 : Valeur de l' augmentation de capital en biens achetés neufs
- 324004 : Valeur de l' augmentation de capital en biens ach. usagés
- 324005 : Valeur de l' augmentation de capital en biens produits par l' ET
- 324006 : Valeur des réductions de capital
- 324007 : Valeur de l' investissement
- 324010 : Valeur prévue pour l' année $x + 1$
- 324011 : Description de l' investissement prévu pour l' année $x + 1$

ces propriétés ont comme caractéristique de mouvement 'situation' et comme dimension de référence 'année'

Notons que chaque entité est caractérisée par un identifiant unique.

CHAPITRE 2 STRUCTURE CONCEPTUELLE DE LA BASE DE DONNEES

Grâce à la réalisation du niveau infologique, nous possédons un schéma de description de la structure des informations (modèle entité-association) . De cette description nous dégagerons les données à stocker dans la base de données et exprimées selon un formalisme relationnel. Nous définirons tout d'abord la notion de relation et montrerons ensuite le passage du modèle 'entité-association', défini au niveau infologique, au modèle relationnel.

2.1 Définition du modèle relationnel

Etant donnés les ensembles A, B, C, \dots on appellera relation n -aire sur A, B, C, \dots tout sous-ensemble R du produit cartésien A, B, C, \dots que l'on écrira $R(A, B, C, \dots) \subseteq A \times B \times C \dots$. R est un ensemble de n -uples de la forme (a, b, c, \dots) , tels que $a \in A, b \in B, c \in C, \dots$

Les ensembles A, B, C, \dots sont appelés les composantes de R . Ils doivent être élémentaires. cfr (BDER p. 176).

Le lecteur trouvera une définition plus complète du modèle relationnel dans (CODD1).

2.2 Passage du modèle 'entité-association' au modèle relationnel

Etant donné une description d'une situation réelle au moyen du modèle 'entité-association', il est possible de définir systématiquement un schéma relationnel de base de données décrivant toutes les données nécessaires à représenter les informations répertoriées au niveau du réel perçu. Ce schéma relationnel sera choisi comme schéma conceptuel car il ne privilégie aucune application particulière et reflète le plus naturellement possible la description du réel perçu réalisée au moyen du modèle 'entité-association'.

Décrivons à présent le procédé systématique de passage du modèle 'entité-association' au modèle relationnel.

- A) A chaque type d'entité, on a associé au niveau infologique, un type de propriété identifiant unique c' est à dire telle que deux entités distinctes de ce type aient des valeurs distinctes de cette propriété.

B) A chaque type d' entité et à chaque type d' association, on associera une ou plusieurs relations de la façon suivante :

B1) Relations contenant les propriétés caractérisées 'événement'

Si un type d' entité ou d' association possède des propriétés caractérisées 'événement', on lui associera une relation ayant comme clé primaire la propriété identifiante de l' entité ou la liste de propriétés identifiantes des entités de l' association et comme composantes non primaires les propriétés classées 'événement'.

B2) Relations contenant les propriétés caractérisées 'situation'

Si un type d' entité ou d' association possède des propriétés caractérisées 'situation', chacune de ces propriétés possède une dimension de référence. Pour chaque dimension de référence correspondant au moins à une propriété, on définira une relation ayant comme clé primaire la propriété identifiante de l' entité (ou la liste de propriétés identifiantes des entités formant l' association). Les propriétés ayant cette dimension de référence constitueront les composantes non primaires et on dira, pour cette raison, que la relation est homogène du point de vue de la dimension de référence. On appellera VERSION d' une telle relation l' ensemble des tuples ayant même valeur de la composante 'dimension de référence'.

B3) Relations contenant les propriétés caractérisées 'existant'

Si un type d' entité ou d' association possède des propriétés caractérisées 'existant', on lui associera une relation ayant comme clé la propriété identifiante de l' entité (ou la liste de propriétés identifiantes des entités de l' association).

Remarques

Le procédé systématique décrit ci-dessus repose sur l' hypothèse qu' une entité ou une association est toujours caractérisée par une et une seule valeur de chacune de ses propriétés. S' il n' en était pas ainsi, on pourrait choisir de représenter la propriété par plusieurs composantes ou par une composante dont les valeurs sont des listes de propriétés. Il faut bien avouer qu' aucune de ces deux solutions n' est réellement satisfaisante, mais

c' est là un défaut inhérent au modèle relationnel (lorsqu' on envisage que les relations soient au moins sous lère forme normale cfr. 4.1.1.).

Après la décomposition des types 'entité-association' en relations, toutes les composantes de celles-ci ont même dimension de référence. Il existe donc deux critères de découpe des types 'entité-association' en relations :

- la caractéristique de mouvement
- la dimension de référence

Après la réalisation de ce niveau nous obtenons la description d' unités homogènes quant à la caractéristique de mouvement et quant à la dimension de référence. Nous obtenons, ainsi, un dictionnaire de données qui servira à l' utilisateur ainsi qu' à l' informaticien.

2.3 Exemples

Reprenons les exemples cités au niveau infologique (cf. 1.3) et examinons comment les entités vont être représentées au niveau conceptuel.

EX 1 : cfr. 1.3 ex 1

L' entité *caractéristiques générales de l' entreprise* sera découpée en deux relations : une première regroupera les propriétés ayant comme caractéristique de mouvement 'existant' et dont les composantes sont les suivantes :

clé : identifiant unique de l' entreprise
 110005 : Identifiant des établissements hors région
 110006 : Adresse de l' EP (siège social)
 110007 : Nom ou raison sociale de l' EP
 110008 : Date de création de EP (publication au Moniteur)
 110009 : Forme juridique de EP
 12000N : Secteur d' activité n de EP

une seconde relation regroupera les propriétés ayant comme caractéristique de mouvement 'situation' (et comme dimension de référence 'trimestre') et dont les composantes sont les suivantes :

clé : identifiant unique de l' entreprise
 440001 : Nombre de mois de retard de paiements vis-à-vis de l' ONSS
 440002 : Montant des paiements en retard à l' ONSS
 440003 : Nombre de mois de retard de paiements au Min.Fin pour pré-compte professionnel
 440004 : Montant des paiements en retard au Min. Fin. pour précompte professionnel

- 440005 : Nombre de mois de retard de paiements au Min. Fin. pour
impôt des sociétés
440006 : Montant des paiements en retard au Min. Fin. pour impôt des
sociétés

EX 2 : cfr. 1.3 ex 2

Il n'y a pas lieu de découper l'entité *capital passif étranger* (*emprunt obligataire*) car toutes ses propriétés ont même caractéristique de mouvement 'événement' (et même dimension de référence 'infinie'). Elle formera une seule relation dont les composantes sont les suivantes :

- clé : identifiant de l'entreprise, identifiant de l'emprunt oblig.
422032 : Montant
422033 : Taux d'intérêt
422034 : Terme
422035 : Date d'émission
422036 : Nombre d'obligations

EX 3 : cfr. 1.3 ex 3

L'entité *capital investi* possédant des propriétés de même caractéristique de mouvement 'événement' formera une seule relation dont les composantes sont les suivantes :

- clé : identifiant de l'établissement, identifiant du type d'investissement
324002 : Valeur de l'augmentation de capital
324003 : Valeur de l'augmentation de capital en biens achetés neufs
324004 : Valeur de l'augmentation de capital en biens ach. usagés
324005 : Valeur de l'augmentation de capital en biens produits par
l'établissement
324006 : Valeur des réductions de capital
324007 : Valeur de l'investissement
324010 : Valeur prévue pour l'année $x + 1$
324011 : Description de l'investissement prévu pour l'année $x + 1$

CHAPITRE 3 IMPLEMENTATION LOGIQUE DE LA BASE DE DONNEES

Après avoir défini, au niveau conceptuel, les relations, définissons à présent les applications possibles sur la base de données. Avant de les détailler, nous définissons les accès logiques permettant de réaliser efficacement les applications.

3.1 Présentation des applications

L' étape normale qui suit la définition du dictionnaire des données consisterait à énumérer un certain nombre d' applications d' aide à la décision. Cette proposition possède un caractère assez ambigu étant donné le choix de l' horizon temporel. Il est impossible, en effet, d' énumérer toutes les applications envisageables à moyen et à long terme. Etant donné le choix d' un 'outil-interface' à structure souple, (BDER) propose de ne pas définir d' applications particulières. C' est pour cette raison que la conception générale du système s' est limitée à deux applications :

- la mise à jour de la base de données
- la consultation de la base de données.

Il est , en effet, impossible de définir toutes les applications impliquées par la politique économique (cfr. BDER p 137).

3.2 Structuration pour l' application consultation

La consultation consistera plus souvent en une extraction d' un lot de données qu' en l' obtention de telle valeur pour tel type de données. Les relations définies en 2.2 sont homogènes quant à la caractéristique de mouvement et quant à la dimension de référence. Elles constituent les unités logiques et cohérentes de consultation car toutes les composantes d' une relation appartiennent au même champ sémantique. Ces unités sémantiques seront appelées pour cette raison RELATIONS LOGIQUES. Ces relations logiques constituent donc des unités homogènes de consultation et contiennent des composantes ayant une forte probabilité d' être demandées en même temps. L' ensemble minimum de composantes qui permet d' identifier de manière unique les occurrences d' une relation est appelé 'identifiant unique' et forme la clé primaire de cette relation (cfr. 2.2). Les autres composantes sont les composantes non primaires de la relation.

A chaque relation définie au niveau conceptuel (cfr. 2.3 ex 1,2,3) correspond une relation logique. En reprenant les exemples précédents nous aurons :

1. la relation logique *identifiant de l' entreprise*
la relation logique *dettes de l' entreprise*
2. la relation logique *capital passif étranger (emprunt obligataire)*
3. la relation logique *capital investi*

3.3 Structuration pour l' application mise à jour

Afin de rendre plus efficaces les opérations de mise à jour, les relations logiques sont découpées en 'sous-relations' : les RELATIONS PHYSIQUES. Tout comme les relations logiques constituent des unités de consultation, les relations physiques, issues de la décomposition des relations logiques, constituent des unités de mise à jour. Les relations logiques peuvent, en effet, contenir des types de données de source ou de fréquence de collecte différente ou des types de données calculées ou non. Examinons les différents critères de découpe des relations logiques en relations physiques.

3.3.1 Critère 'source de données'

Il apparaît évident que des types de données collectés par des organismes-sources différents ne peuvent appartenir à la même relation physique, sous peine de s' obliger à de multiples manipulations lors de la mise à jour. Ainsi, toute relation physique devient homogène du point de vue de la source de collecte. Une relation logique sera donc scindée en autant de relations physiques qu' il y a de composantes concernées par des organismes-sources différents.

3.3.2 Critère 'fréquence de collecte'

L' opération de mise à jour consiste, la plupart du temps, en l' apport de nouvelles 'versions' à l' entrepôt. En effet, elle ne consiste pas à remplacer une série de données collectées à une date précise par une autre série de données collectées à une autre date, mais bien à constituer des séries de données collectées à différentes dates : l' utilisateur aura, ainsi, accès à plusieurs versions d' une série de données. Les données d' une relation logique ayant même dimension de référence peuvent être collectées à des moments différents c' est à dire qu' elles n' auront pas la même fréquence de collecte.

3.3.3 Critère 'donnée calculée'

Dans les relations logiques possédant une caractéristique de mouvement 'situation', c' est à dire stables durant une période de temps, certaines des composantes sont calculées sur base d' autres composantes appartenant à la base de données. Afin d' interdire facilement et rapidement la mise à jour de ces composantes calculées, nous avons pris en considération ce critère pour effectuer la décomposition de relations logiques en relations physiques.

3.3.4 Autres critères

Il existe d' autres critères de décomposition des relations logiques qui ne sont pas pris en considération pour la structuration conceptuelle de l' entrepôt ni pour la consultation, ni pour la mise à jour. Il s' agit des critères de 'possibilité de saisie' et de 'possibilité de divulgation'. Le critère 'possibilité de saisie' est relatif à la disponibilité actuelle d' un type de données dans un organisme déterminé. Le critère de 'possibilité de divulgation' est relatif à l' autorisation de consultation d' un type de données.

3.3.5 Exemples

Reprenons les exemples des chapitres précédents :

EX 1 : la relation logique *identifiant de l' entreprise* ne sera pas découpée en relations physiques car toutes ses données ont la même source (registre de commerce) et la même fréquence de collecte (sans) et ne sont pas calculées. Il s' agit de la relation numero 2 (cfr. annexe 1) et dont la description est la suivante :

clé : identifiant de l' entreprise
 110005 : Identifiant des établissements hors région
 110006 : Adresse de EP (siège social)
 110007 : Nom ou raison sociale de l' EP
 110008 : Date de création de l' EP (publication au moniteur)
 110009 : Forme juridique de l' EP
 12000N : Secteur d' activité n de EP

la relation logique *dettes de l' entreprise* sera découpée selon le critère de source et de fréquence de collecte pour aboutir à la définition de trois relation physiques. Les descriptions de ces relations (3.1, 3.2, 3.3 cfr. annexe 1) sont les suivantes :

- clé : identifiant de l' entreprise

440001 : Nombre de mois de retard de paiements vis-à-vis de l' ONSS

440002 : Montant des paiements en retard vis-à-vis de l' ONSS

La relation physique ci-dessus a comme source l' ONSS

et comme fréquence de collecte le trimestre

- clé : identifiant de l' entreprise

440003 : Nombre de mois de retard de paiements au Min. Fin. pour
précompte professionnel

440004 : Montant des paiements en retard vis-à-vis du Min. Fin.
précompte professionnel

La relation physique ci-dessus a comme source le Ministère Finances

et comme fréquence de collecte le trimestre

- clé : identifiant de l' entreprise

440005 : Nombre de mois de retard de paiements au Min. Fin. pour
impôt des sociétés

440006 : Montant des paiements en retard vis-à-vis du Min. Fin.
impôt des sociétés

La relation physique ci-dessus a comme source le Ministère Finances

et comme fréquence de collecte le trimestre

EX 2 : la relation logique *capital passif étranger (emprunt obligataire)* ne

sera pas découpée en relations physiques car toutes ses données ont la même source (Ministère des Finances) et la même fréquence de collecte (variable) et ne sont pas calculées. Cette relation porte le numero 12 dans l' annexe 1 et sa description est la suivante :

clé : identifiant de l' entreprise, identifiant de l' emprunt oblig.

422032 : Montant

422033 : Taux d' intérêt

422034 : Terme

422035 : Date d' émission

422036 : Nombre d' obligations

EX 3 : la relation logique *capital investi* après la décomposition selon les

critères de source, de fréquence de collecte et de calcul permet

d' obtenir la définition de trois relations physiques dont les numéros sont dans l' annexe 1 et dont les descriptions

sont les suivantes :

- La relation physique ci-dessous a comme source INS (Institut National de Statistique) et comme fréquence de collecte l' année et n' a pas de composantes calculées.

clé : identifiant de l' établissement, identifiant du type d' inv.

324002 : Valeur de l' augmentation de capital

324003 : Valeur de l' augmentation de capital en biens achetés neufs

324004 : Valeur de l' augmentation de capital en biens ach. usagés

324005 : Valeur de l' augmentation de capital en biens produits par
l' entreprise

324006 : Valeur des réductions de capital

- La relation physique ci-dessous a comme source INS,
comme fréquence de collecte l' année
et n' a que des composantes calculées.

clé : identifiant de l' établissement, identifiant du type d' inv.

324007 : Valeur de l' investissement

- La relation physique ci-dessous a comme source INS,
comme fréquence de collecte 'inex'
et n' a pas de composantes calculées.

clé : identifiant de l' établissement, identifiant du type d' inv.

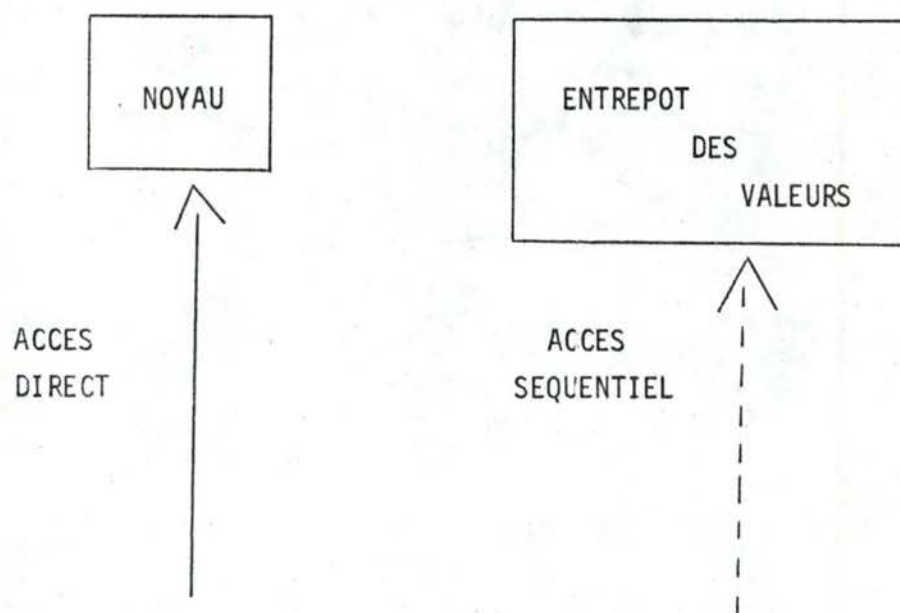
324010 : Valeur prévue pour l' année $x + 1$

324011 : Description de l' investissement prévu pour l' année $x + 1$

Notons que, fréquence de collecte 'sans' signifie que la fréquence de collecte est très faible; c' est le cas pour les données d' identification, fréquence de collecte 'inex' (inexistante) implique que ces données ne sont pas collectées à l' heure actuelle, lorsque la collecte de données se fait chaque fois qu' une nouvelle occurrence apparait celle-ci aura une fréquence de collecte 'var' (variable).

3.4 Accès logiques

Il est généralement admis (cfr. CBDH) que le niveau de description logique prend en charge les accès à implémenter pour réaliser efficacement les applications. La fonction essentielle de la base de données étant d'assurer, en tant qu'outil-interface, des entrées et des sorties de grosses quantités de données avec le minimum de transformation, il est évident que des accès séquentiels s'imposent. L'ensemble des relations accédées séquentiellement sera appelé ENTREPOT DES VALEURS. Il est cependant important de privilégier certaines relations de la base de données en les implémentant en accès direct. L'ensemble de ces dernières constituera le NOYAU de la base de données et contiendra par exemple des données signalétiques utilisées fréquemment pour un nombre limité d'occurrences. La base de données est, par conséquent, composée de l'entrepôt des valeurs et du noyau.



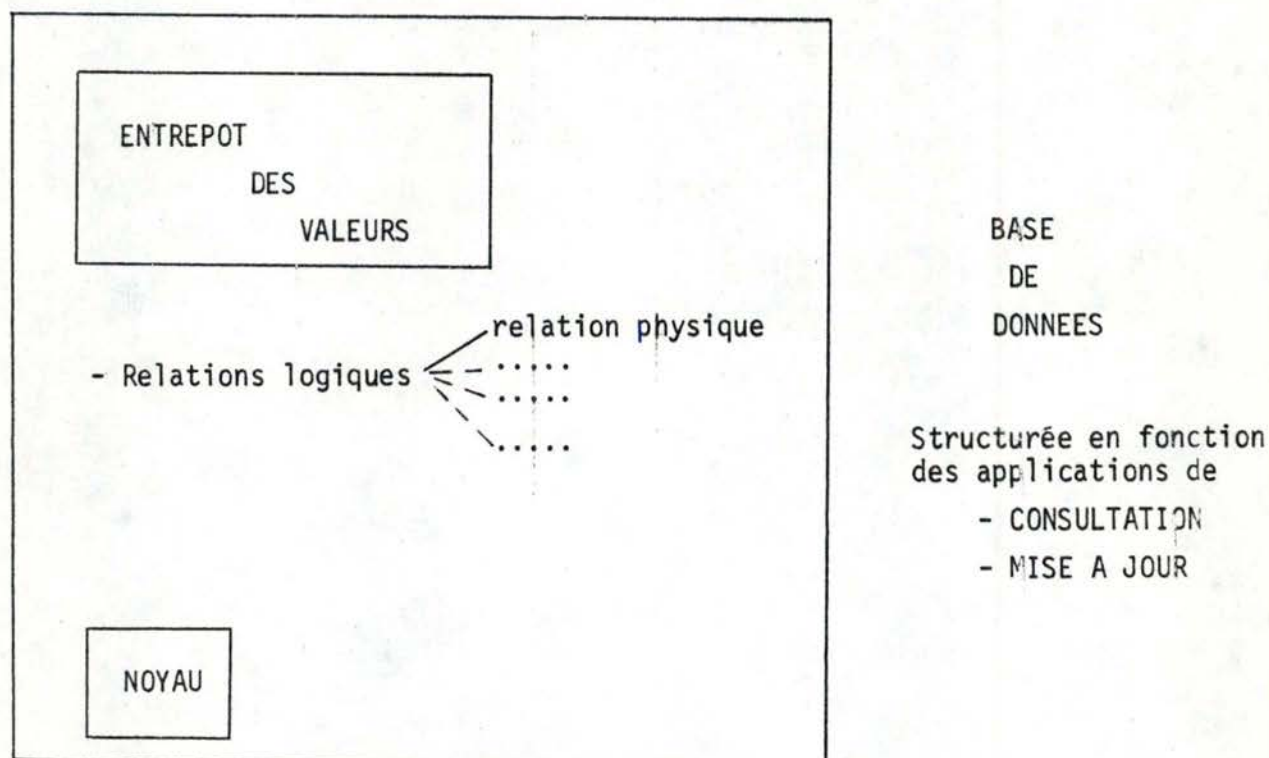
- figure 1 -

3.5 Schéma de la base de données

Les relations logiques sont des unités de consultation homogènes
du point de vue de la - caractéristique de mouvement
- dimension de référence

Les relations logiques sont composées d' une ou plusieurs relations
physiques. Les relations physiques sont des unités de mise à jour homogènes
du point de vue du -critère 'source des données'

- critère 'fréquence de collecte'
- critère 'donnée calculée'.



- figure 2 -

3.6 Liste et brève description des opérations des 2 applications

L' utilisateur désirant exécuter ces applications doit évidemment être reconnu comme étant une personne ayant l' autorisation de manipuler les données.

Pour l' énumération des opérations des deux applications, nous nous sommes inspirés des propositions de (BDER pp. 216-230).

a) L' application consultation permet trois types d' opération :

- la consultation du dictionnaire de données
- la consultation du noyau
- la consultation de l' entrepôt des valeurs.

Ces opérations pourront être effectuées à la fois par l' utilisateur (pouvoirs publics ou entreprises manufacturières) et le gestionnaire de la base de données.

Les différentes opérations de consultation consistent soit en un listage d' une relation logique, soit en une combinaison de commandes d' interrogation de base de données telles que :

- projection d' une relation sur une liste de composantes
- sélection sur filtre
- jointure de deux relations sur une liste de composantes
- adjonction de nouvelles composantes numériques à une relation.

b) L' application mise à jour permet quatre types d' opérations :

- la mise à jour massive de l' entrepôt des valeurs
- la mise à jour ponctuelle du noyau
- la mise à jour ponctuelle de l' entrepôt des valeurs
- la mise à jour ponctuelle du dictionnaire de données.

Seule la mise à jour massive de l' entrepôt des valeurs, consistant en l' apport de nouvelles versions de données à une relation physique, pourra être effectuée par l' utilisateur; ceci afin d' assurer la cohérence de la base de données. Les autres opérations de mise à jour seront seules permises au gestionnaire de la base de données.

Rappelons que les relations physiques contenant les composantes calculées sont mise à jour uniquement par l' intermédiaire de programmes de calcul.

3.6.1 Description de l' application consultation

L' application de consultation a pour entité de base la relation logique qui regroupe des types de données susceptibles, en raison de leur sémantique, d' être demandées en même temps.

Cette opération permet trois types d' opération :

- la consultation du dictionnaire de données
- la consultation du noyau
- la consultation de l' entrepôt des valeurs.

Chacune de ces opérations peut consister, soit en une demande d' une relation logique telle qu' elle existe dans la base de données, soit en un groupement de commandes d' interrogation de base de données dont nous donnerons la notation, la définition et l' utilité.

3.6.1.1 Listage d' une relation logique

Le listage d' une relation logique consiste à éditer toutes les données d' une relation logique présentes dans la base de données. Deux possibilités s' offrent à l' utilisateur : la première présente une vue diachronique des données en les regroupant par valeur de la clé primaire (argument primaire) et par valeur de la version (argument secondaire). Quant à la deuxième possibilité, elle permet la vue synchronique des données en inversant l' ordre des arguments. Les données sont alors triées par valeur de la version (argument primaire) et par valeur de la clé (argument secondaire).

Le format général de la commande sera : LIST nom de relation logique

3.6.1.2 Projection d' une relation sur une liste de composantes (1)

a) notation : PROJ (relation (liste de composantes))

b) définition : PROJ (R (L)) $\equiv \{ l \mid \exists t \in C_L^P : (l, t) \in R \}$

où C_L^P = complémentaire de L par rapport à P

R = nom d' une relation logique

L = liste des composantes sur laquelle est projetée R

l = occurrence de L, occurrence du résultat

P = liste des composantes de R

t = occurrence de (P - L)

Le résultat de l' opération sera l' ensemble des occurrences l.

- c) utilité : en projetant une relation sur une liste de composantes, on élimine des composantes de la relation n' appartenant pas à cette liste.

Lors d' une projection, il sera possible d' éliminer les duplicatas des occurrences du résultat

3.6.1.3 Sélection sur filtre (1)

- a) notation : SEL (relation { composante op valeur })

- b) définition : SEL (R { L op v }) \equiv

$$\{ (t,w) \mid \exists t \in \complement_L^P \wedge w \text{ op } v : (t,w) \in R \}$$

où \complement_L^P = complémentaire de L par rapport à P

R = nom d' une relation logique

L = composante de R

op = opérateur de comparaison (< ≤ = ≥ > ≠)

{L op v} = filtre

v = expression pouvant par exemple prendre la forme suivante :

$$v = v1 \text{ ou } v2$$

$$v = v1 \text{ et } v2$$

...

w = occurrence de L

P = liste des composantes de R

t = occurrence de (P-L)

(t,w) = occurrence du résultat

- c) utilité : la sélection permet de choisir les occurrences d' une relation logique à partir d' un filtre défini sur une liste de composantes de cette relation.

3.6.1.4 Jointure de deux relations (1)

- a) notation : JOIN (relation (liste de composantes) *
relation (liste de composantes))

- b) définition : JOIN (R1 (L1) * R2 (L2)) \equiv

$$\{ ((l1=l2), t1, t2) \mid \exists t1 \in \complement_{L1}^{P1}, t2 \in \complement_{L2}^{P2} (l1,t1) \in R1 \wedge (l2,t2) \in R2 \wedge l1=l2 \}$$

où \complement_{Li}^{Pi} = complémentaire de Li par rapport à Pi

R1, R2 = nom d' une relation logique

L1 = liste de composantes de R1 intervenant dans la jointure

L2 = liste de composantes de R2 intervenant dans la jointure

P1 = liste des composantes de R1

P2 = liste des composantes de R2

t1 = occurrence de (P1-L1)

t2 = occurrence de (P2-L2)

l1 = occurrence de L1

l2 = occurrence de L2

$((l1=l2), t1, t2)$ = occurrence du résultat

- c) utilité : la jointure de deux relations permet de ramener dans une relation les composantes de ces deux relations (logiques).

3.6.1.5 Adjonction de nouvelles composantes (1)

Selon une proposition de (BDER) , un utilisateur peut ajouter, à une relation logique existante, une ou plusieurs composantes de type numérique dont la valeur est une fonction arithmétique des valeurs d' autres composantes de cette relation. Nous avons estimé inutile de définir cette opération car elle nous semble contraire à l' esprit de (BDER), c' est à dire de ne pas rendre des services particuliers à l' utilisateur. La base de données fournira, grâce à la consultation, toutes les données d' une relation logique que l' utilisateur pourra manipuler à sa guise pour ajouter une nouvelle composante

3.6.2 Description de l' application mise à jour

La mise à jour de l' entrepôt consiste principalement en l' apport de nouvelles données, sans pour autant rendre inaccessible les anciennes valeurs. L' unité fondamentale manipulée par cette application est la relation physique, dont toutes les composantes ont la même source et fréquence de collecte ainsi que le même critère de calcul; grâce à cette homogénéité, les données d' une telle relation sont mise à jour au même moment par le même utilisateur. Ce sont des données non calculées.

L' application de mise à jour permet quatre types d' opération :

- la mise à jour massive de l' entrepôt des valeurs
- la mise à jour ponctuelle du noyau
- la mise à jour ponctuelle du dictionnaire de données
- la mise à jour ponctuelle de l' entrepôt des valeurs.

Les trois dernières opérations relèvent seulement de la compétence du gestionnaire de la base de données, ceci afin d' assurer la cohérence de la base de données.

3.6.2.1 Mise à jour massive de l' entrepôt des valeurs

La mise à jour massive de l' entrepôt des valeurs consiste en l' apport de nouvelles données (non calculées). Cette opération est permise à tout utilisateur autorisé.

3.6.2.2 Mise à jour ponctuelle du noyau

La mise à jour du noyau est une opération occasionnelle pouvant entraîner l' ajoute, la modification ou la suppression de valeurs dans le noyau. Le gestionnaire de la base de données est seul à pouvoir effectuer cette opération.

3.6.2.3 Mise à jour ponctuelle de l' entrepôt

Des modifications, des ajoutes et des suppressions peuvent, le cas échéant, être effectuées par le gestionnaire de la base de données afin d' assurer une meilleure 'qualité' des données par exemple , la correction d' erreur,...

3.6.2.4 Mise à jour ponctuelle du dictionnaire de données

Des modifications de structure peuvent entraîner la modification des données du dictionnaire. Celles-ci, effectuées par le gestionnaire de la base de données, doivent être rendues possibles afin d'assurer un suivi parfait des modifications de la structure du système.

Ce type d'opération comprend les commandes suivantes :

- l'ajoute d'une relation logique
- la suppression d'une relation logique
- l'ajoute d'une relation physique
- la suppression d'une relation physique
- l'ajoute d'une composante à une relation physique existante
- la suppression d'une composante d'une relation physique existante.

Ceci permet d'assurer l'extensibilité de la base de données cfr. 5.5.2

CHAPITRE 4 PROBLEMES POSES PAR LA STRUCTURATION DE L' ENTREPOT

Il nous a semblé utile d' apporter quelques précisions à l' approche informatique définie dans les chapitres 4 et 5 de (BDER pp. 167-263). Certaines ambiguïtés nous paraissaient y subsister. Nous examinerons les questions suivantes :

- la mise sous 3ème forme normale de toutes les relations de la base de données
- précisions quant aux critères de structuration des données
- différenciation des notions de dimension de référence et de fréquence de collecte
- restructuration de certaines relations de l' entrepôt pour des raisons diverses

4.1 Normalisation des relations

Afin d' éliminer toute redondance de données et d' éviter des modifications en cascade, E.F. Codd (CODD2) propose la normalisation des relations. Si un des deux arguments précités joue un rôle important, la mise sous 3ème forme normale est indispensable.

Pour pouvoir la réaliser, on suit la démarche suivante :

- donner la liste des dépendances fonctionnelles
- en éliminer celles qui peuvent être déduites d' autres dépendances et qui sont redondantes c' est à dire qui n' apportent aucune information supplémentaire.

Cependant, pour des raisons de simplicité, nous préférons définir les différentes formes normales en montrant l' intérêt de la 2ème forme normale, car celle-ci nous intéresse particulièrement avant de procéder à la découpe des relations à normaliser.

4.1.1 Définition des trois formes normales et de certains mots clés

Pour ces définitions nous nous sommes basés sur (CBDH)

1ère forme normale : toutes les composantes d' une relation sont des composantes simples et non des relations.

2ème forme normale : implique la 1ère forme normale; en outre, toutes les composantes non primaires dépendent totalement de la clé primaire et non d' une partie de la clé seulement.

3ème forme normale : implique la 2ème forme normale; de plus il n'y a pas de dépendance fonctionnelle transitive dans les composantes non primaires.

Dépendance fonctionnelle : dans une relation, une composante dépend fonctionnellement d'une autre si à chaque occurrence de la 2ème composante ne correspond qu'une seule occurrence de la 1ère composante.

Soit $R (A, B, C, D)$

où R = relation

A, B, C, D = composantes de R

D dépend fonctionnellement de B dans R noté $B \xrightarrow{R} D \equiv$

$$\forall b \in \text{PROJ}(R(B)), \exists ! d \in D : (b, d) \in \text{PROJ}(R(B, D))$$

où b, d = occurrences respectivement de B et D

Dépendance totale : Dans une relation, une composante dépend totalement d'un ensemble de composantes si elle ne dépend fonctionnellement d'aucun de ses sous-ensembles mais bien de l'ensemble.

Soit $R (L)$

où R = relation

L = liste des composantes de R

Soit $I \subseteq L$

Soit $J \subseteq L (I \neq J)$

notons $I \xrightarrow[T]{R} J$ la dépendance totale de I vers $J \equiv$

$$(\nexists I' \subset I : I' \xrightarrow{R} J) \text{ et } I \rightarrow J$$

Dépendance transitive : Soit R une relation

A, B, C des composantes de R

A dépend transitivement de $C \iff$ il existe une composante B telle que

$$A \xrightarrow{R} B \text{ et } B \xrightarrow{R} C$$

Il est alors évident qu'on a $A \xrightarrow{R} C$

4.1.2 Interêt de la 2ème forme normale

Quelques relations de l' entrepôt ne sont pas sous 3ème forme normale, parce qu' elles ne sont pas sous 2ème forme normale. C' est pourquoi nous montrerons l' utilité de la 2ème forme normale à l' aide d' un exemple pour lequel nous utiliserons la relation 35.3 de (BDER p. 194) intitulée *charge d' utilisation de la réserve de potentiel humain* et dont la description est la suivante :

clé : nom de l' établissement (C1), qualification d' emploi (C2)
 312009 : Valeur globale des rémunérations brutes annuelles du personnel
 312010 : Salaire annuel brut
 312011 : Supplément par année d' ancienneté
 312012 : Système de rémunération habituel (R_i)

On pourrait avoir les réalisations suivantes :

R (C1, C2, 312009, 312010, 312011, 312012)

ET1	Q1	2400000	150000	10000	S1
ET1	Q1	2400000	300000	4000	S1
ET2	Q1	5000000	200000	2000	S2
...

Si une occurrence de 312012 (S1 par exemple) change pour un établissement (C1), il faut, alors, modifier un certain nombre de lignes (1ère et 2ème ici). Par contre, en éclatant cette relation en deux relations R1 (C1, 312009, 312012) et R2 (C1, C2, 312010, 312011) avec comme réalisations :

pour R1	ET1	2400000	S1	pour R2	ET1	Q1	150000	10000
	ET2	5000000	S2		ET1	Q2	300000	4000
					ET2	Q1	200000	2000

il suffirait de modifier une ligne de R1 (1ère ici), grâce au fait que toutes les composantes non primaires dépendent désormais totalement de leur clé primaire.

La mise sous 2ème forme normale ne facilite donc pas seulement la modification, mais élimine aussi les données redondantes dans les relations. Dans notre exemple, la valeur de S1 se retrouve dans deux occurrences de R, alors qu' une seule présence suffirait.

En ce qui concerne la mise à jour, elle consiste principalement en l' apport de nouvelles données à la fin d' une relation physique. Pour cette raison, le principal argument en faveur de la normalisation est le gain de place assuré par l' élimination de toute redondance au sein d' une relation.

4.1.3 Mise sous 3ème forme normale des relations

Pour qu'une relation soit sous 3ème forme normale, elle doit évidemment être sous 2ème forme normale. Or dans le subset de données proposé, il subsiste trois relations physiques qui ne sont pas sous 2ème forme normale. Pour chacune de ces relations, nous avons éliminé des composantes avec lesquelles nous avons créé une nouvelle relation logique. Exposons l'ancienne et la nouvelle description de ces relations.

4.1.3.1 Relation physique *aide à l'investissement déjà attribuée*

Cette relation a le numero 26.1 dans (BDER pp. 190-191) et possède la description suivante :

clé : identifiant de l'entreprise

- 414002 : Identifiant des EP ayant obtenu une aide à l'inv. de type 1
- 414003 : EP ayant demandé une aide à l'inv. de type 1 est-elle PME ?
- 414004 : Numero dossier demande d'aide à l'investissement
- * 414005 : Valeur totale des investissements à aider
- * 414006 : Nature des investissements à aider
- 414008 : Bénéficiaire de l'aide à l'investissement
- 414009 : But de l'investissement
- 414010 : Date de mise en place d'un effectif min. suite à l'aide à l'investissement
- 414011 : Nombre min. d'emplois imposés lors de l'aide à l'inv.
- 414012 : Date de l'achèvement du programme d'inv.
- 414013 : Nombre d'emplois créés après le programme d'inv.
- 414014 : Date du début de l'octroi d'aide à l'investissement
- 414015 : Date de clôture de l'octroi d'aide à l'inv.
- * 414016 : Interêt nominal demandé à l'EP par l'organisme prêteur
- * 414017 : Part de l'intérêt payé par l'Etat (inv. type 1)
- * 414018 : Valeur globale des subsides (inv. de type 1)
- * 414019 : Valeur globale des subsides (inv. de type 2)

Certaines composantes (414005, 414006, 414016, 414017, 414018, 414019) (précédées par '*') ont comme clé primaire le numéro d'entreprise (ET), tandis que les autres dépendent de deux composantes : numéro de l'entreprise (ET) et id. du type d'investissement (Tyinv).

Nous avons, donc, redéfini la relation et ajouté une nouvelle relation logique. La relation 26.1 a été éclatée en deux relations qui portent les numéros 26.1 et 29 dans l'annexe 1 et dont les descriptions sont les suivantes :

Relation 26.1 *aide à l'investissement déjà attribuée*

clé : identifiant de l'entreprise

- * 414005 : Valeur totale des investissements à aider
- * 414006 : Nature des investissements à aider

- * 414016 : Interêt nominal demandé à l' EP par l' organisme prêteur
- * 414017 : Part de l' intérêt payé par l' Etat (inv. type 1)
- * 414018 : Valeur globale des subsides (inv. de type 1)
- * 414019 : Valeur globale des subsides (inv. de type 2)

Relation 29 *aide à l' investissement déjà attribuée*

- clé : identifiant de l' entreprise, id. du type d' aide à l' invest.
- 414002 : Identifiant des EP ayant obtenu une aide à l' inv. de type 1
 - 414003 : EP ayant demandé une aide à l' inv. de type 1 est-elle PME ?
 - 414004 : Numéro dossier demande d' aide à l' inv.
 - 414008 : Bénéficiaire de l' aide à l' inv.
 - 414009 : But de l' investissement
 - 414010 : Date de mise en place d' un effectif min. suite à l' aide à l' investissement
 - 414011 : Nombre min. d' emplois imposés lors de l' aide à l' inv.
 - 414012 : Date de l' achèvement du progr. d' inv.
 - 414013 : Nombre d' emplois créés après le progr. d' inv.
 - 414014 : Date de début de l' octroi d' aide à l' inv.
 - 414015 : Date de clôture de l' octroi d' aide à l' inv.

4.1.3.2 Relation physique *charge d' utilisation de potentiel humain*

La description de cette relation se trouve en 4.1.2. Rappelons que les données 312009 et 312012 ne dépendent que d' une partie de la clé c' est à dire identifiant de l' établissement, alors que les deux autres composantes (312010 et 312011) dépendent totalement de la clé c' est à dire identifiant de l' établissement et qualification d' emploi.

La relation 35.3 perd, donc, deux composantes avec lesquelles une nouvelle relation logique est créée et dont les descriptions sont les suivantes :

Relation numéro 39.3 (nouveau numéro) dans l' annexe 1

clé : identifiant de l' établissement

- 312009 : Valeur globale des rémunérations brutes annuelles du pers.
- 312012 : Système de rémunération habituel

Relation numéro 40 (nouveau numéro) dans l' annexe 1

clé : identifiant de l' établissement, id. qualification d' emploi

- 312010 : Salaire brut annuel
- 312011 : Supplément par année d' ancienneté

4.1.3.3 Relation physique *volume des ventes*

Cette relation porte le numéro 49.1 dans (BDER) et dont la description est la suivante :

clé : identifiant de l' établissement, catégorie de produit

- 213001 : Valeur des ventes ou travaux au cours de l' exercice
- 213004 : Valeur des ventes destinées à l' étranger
- * 213005 : Valeur des ventes de produits sortis sans transformation
- 213006 : Quantités vendues au cours de l' exercice
- 213009 : Quantités vendues à l' étranger
- * 213014 : Valeur des produits cédés à d' autres ET de l' EP
- * 213015 : Valeur des travaux effectués pour des tiers
- * 213006 : Valeur des produits accessoires

Nous nous trouvons en présence d' une part, de données (213001, 213004, 213006, 213009) dépendant de la clé (identifiant de l' établissement, cat. de produit) et d' autre part, de données (*) dépendant d' une partie de la clé. Nous avons redéfini cette relation en éliminant quatre composantes et créé une relation logique dont les descriptions sont les suivantes :

Relation 54.1 (nouveau numéro dans l' annexe 1) *volume des ventes*

clé : identifiant de l' établissement, catégorie de produit

- 213001 : Valeur des ventes ou travaux au cours de l' exercice
- 213004 : Valeur des ventes destinées à l' étranger
- 213006 : Quantités vendues au cours de l' exercice
- 213009 : Quantités vendues à l' étranger

Relation 55 (numero dans l' annexe 1) *volume des ventes*

- * 213005 : Valeur des ventes de produits sortis sans transformation
- * 213014 : Valeur des produits cédés à d' autres ET de l' EP
- * 213015 : Valeur des travaux effectués pour des tiers
- * 213016 : Valeur des produits accessoires

4.1.3.4 Remarque

Le respect rigoureux de la démarche à suivre pour la structuration d' une base de données (cfr. introduction) aurait entraîné la mise de toutes les relations sous 3ème forme normale.

Au niveau infologique a été défini un schéma de description des informations dans le modèle 'entité-association' dont les principaux éléments sont l' entité, l' association et la propriété. Dans ce modèle, une propriété appartenant à une entité ou à une association est une qualité que les individus attribuent à cette entité ou à cette association. Nous pouvons, donc, dire que d' une part l' entité, et d' autre part les entités regroupées au

sein d' une association caractérisées par leurs propriétés sont des données identifiantes, ce qui n' a pas été rigoureusement respecté dans la démarche suivie par (BDER). Si la structuration du réel perçu a été correctement réalisée grâce au modèle 'entité-association', toutes les relations doivent être sous troisième forme normale car elles dépendent totalement de la clé primaire. En effet, aux données identifiantes, définies au niveau infologique, vont toujours correspondre des clés primaires dans les relations logiques et physiques qui, en conséquence, devraient automatiquement se trouver toutes sous 3ème F. N..

4.2. Précisions quant aux critères de structuration des données

Rappelons d' une part, que deux critères de scission de données sont appliqués lors du passage du niveau infologique au niveau conceptuel : ce sont les critères de dimension de référence et de mouvement, et, d' autre part, que trois autres critères permettent la découpe des relations logiques en relations physiques : ce sont les critères de source, fréquence de collecte et de calcul.

Deux parmi ces cinq critères, les critères de mouvement et de calcul, n' ont pas été utilisés comme critères de structuration des données dans la démarche suivie par (BDER). La justification de l' utilisation de ces deux critères est la suivante :

critère de mouvement : le critère de dimension de référence ne s' appliquant
----- qu' après l' utilisation du critère de mouvement et seulement aux données de type 'situation', il nous a semblé intéressant de faire une découpe préliminaire afin de pouvoir différencier les données stables dans le temps (événement), les données peu soumises à des modifications (existant) et des données stables durant une période de temps (situation). Dans le cas contraire, les données de type 'événement' et de type 'existant' n' auraient pas pu être distinguées.

critère de calcul : nous devons distinguer les relations physiques auxquelles
----- l' utilisateur a le droit d' apporter de nouvelles données, des relations physiques contenant uniquement des données calculées et qu' il lui est interdit de mettre à jour.

4.3 Dimension de référence et fréquence de collecte

Afin d'écarter toute ambiguïté concernant les deux concepts de dimension de référence et de fréquence de collecte, rappelons leur définition:

Dimension de référence : toutes les données d'une relation logique et par conséquent celles de ses relations physiques ont aussi même dimension de référence : celle-ci détermine donc, les caractéristiques temporelles des relations cfr. (BDER p. 180).

Fréquence de collecte : toutes les données d'une relation physique, mais pas nécessairement toutes celles de toutes les relations physiques d'une même relation logique, ont même fréquence de collecte cfr. (BDER pp. 181-182); cette caractéristique est d'ailleurs un des trois critères de scission des relations logiques en relations physiques.

Supposons que la collecte de certaines données d'une relation logique se fasse trimestriellement, tandis que celle d'autres données de la même relation logique se fait annuellement. Si leur dimension de référence est trimestrielle, on ajoutera tous les ans quatre versions en même temps pour les données de fréquence de collecte annuelle. La fréquence de collecte ne spécifie que le moment où l'on collecte la donnée.

Il nous semble que, théoriquement, la fréquence de collecte peut être plus grande, plus petite ou égale à la dimension de référence ce qui est contraire à ce que (BDER) propose p. 218 : dimension de ref. \geq fréq. de collecte.

4.4 Modifications dans la structuration de l'entrepôt

Nous voudrions proposer quelques modifications ponctuelles à effectuer dans l'entrepôt et essayer de compléter le contenu du tableau des relations présentées dans (BDER pp. 185-201)

4.4.1 Ajoute de deux relations à l'entrepôt

Nous avons ajouté à l'entrepôt des données concernant l'indice du prix de détail ainsi que celles concernant certains impôts indirects car elles sont utilisées dans des calculs mais absentes de la base de données.

4.4.1.1 Relation *indice du prix de détail* (n° 67 annexe 1)

Nous avons créé une relation qui a comme seule composante non primaire l' indice du prix de détail (type de donnée n° 111101).

Ce dernier intervient, en effet, comme opérande dans le calcul de deux types de données (n° 424005 et n° 424007) représentant des valeurs tenant compte du coût de la vie, alors qu' il était absent du dictionnaire.

Voici la description de la nouvelle relation :

111101 : indice du prix de détail

ainsi que celle des deux formules de calcul en cause :

Formule PC43 (cfr. annexe 3) : $424005 = \frac{424001}{111101}$

où 424005 : Valeur des dividendes, compte tenu du coût de la vie
424001 : Valeur des dividendes

Formule PC41 (cfr. annexe 3) : $424007 = \frac{312005}{111101}$

où 424007 : Valeur de la masse salariale à prix constant
312005 : Valeur globale des dépenses en personnel

4.4.1.2 Relation *impôts indirects* (n° 68 annexe 1)

Le type de donnée numero 432012 (valeur des autres impôts indirects), opérande dans la formule de calcul (cfr. PC46 annexe) est absent de l' entrepôt. Nous avons donc ajouté une relation concernant l' imposition et qui comporte quatre composantes non primaires. Nous avons pu regrouper ces composantes car elles satisfont aux conditions décrites au niveau de l' implémentation logique pour la construction des relations physiques. La description de cette relation est la suivante :

clé : identifiant de l' établissement

432009 : Montant annuel global de la TVA déductible sur les achats
de combustibles, de matières premières
432010 : Valeur annuelle de la TVA perçue sur les ventes
432011 : Valeur de la TVA versée à l' Etat
432012 : Valeur des autres impôts indirects

La formule de calcul concernée (PC46) est la suivante :

$210003 = 210002 + 423007 - 432012$

où 210003 : Valeur ajoutée brute au coût des facteurs
210002 : Valeur ajoutée brute au prix du marché
423007 : Valeur de l' excédent brut de l' exploitation
432012 : Valeur des autres impôts indirects

4.4.2 Modification d' une formule de calcul

Nous avons corrigé la formule de calcul (cfr. BDER p. 101)

$424007 = 312002 / 111101$ en

$424007 = 312005 / 111101$ (cfr. annexe 3 PC41)

où :

424007 : Valeur de la masse salariale à prix constants

312002 : Valeurs des cotisations patronales pour les trav. intellectuels

111101 : indice des prix de détail

312005 : Valeur globale des dépenses en personnel

Si la formule était maintenue, alors le type de données n° 424007 aurait être intitulé *valeur de la masse salariale intellectuelle à prix constants*.

4.4.3 Etablissement / entreprise cfr. (BDER p. 77 note 1)

Prenons comme exemple la relation n° 54 (cfr. BDER) intitulée *structure du prix de revient global* et décrite comme suit :

clé : identifiant de l' établissement , id. de l' entreprise

221101 : Prix de revient pour l' ensemble de la production

221102 : Prix de revient pour l' ens. de la prod. en matières prem.

221103 : Prix de revient pour l' ensemble de la production en mat. accessoires et de conditionnement

221104 : Prix de revient de l' ensemble de la production en salaires, appointements et charges sociales

221105 : Prix de revient pour l' ens. de la production en énergie

221106 : Prix de revient pour l' ens. de la prod. en amortissements

221107 : Prix de revient pour l' ens. de la prod. en frais généraux

221108 : Prix de revient pour l' ens. de la prod. en frais de vente

221109 : Prix de revient pour l' ens. de la prod. en divers

Cette relation a comme clé soit id. de l' établissement, soit id. de l' entreprise, soit les deux. Il n' est cependant pas nécessaire de mémoriser pour tous les établissements l' entreprise à laquelle ils appartiennent. Cette information peut être facilement retrouvée grâce à la consultation de la relation 1.1 intitulée *identifiant de l' établissement* (cfr. annexe 1) dont la description est la suivante :

clé : identifiant de l' établissement

110001 : Identifiant de ET, son nom

110002 : Adresse de ET

110003 : Date de création ET, date de mise en activité

110004 : Identifiant de EP dont dépend ET

12000N : Secteur d' activité N de ET

Il sera ensuite facile de cumuler les valeurs trouvées pour chaque établissements d' une entreprise pour trouver la valeur du prix de revient désirée pour celle-ci.

CHAPITRE 5 IMPLEMENTATION PHYSIQUE DE LA BASE DE DONNEES

Après avoir décrit brièvement les trois premiers niveaux de structuration des données (niveau infologique, conceptuel et logique) abordons à présent le quatrième et dernier niveau, celui de l'implémentation physique de la base de données. Nous utiliserons une base de données descriptive, définie comme système de gestion de la base de données dans (BDER), pour gérer les données.

Avant de décrire les deux applications définies sur la base de données : la consultation et la mise à jour, nous ferons quelques remarques préliminaires à l'implémentation physique c'est à dire la description des modifications que nous proposons d'apporter à la base de données descriptive (B.D.D.), l'incorporation de la notion de 'version' aux relations et la réalisation d'une correspondance biunivoque entre une relation physique et un fichier. Nous décrivons, ensuite, le principe général de la création de la base de données ainsi que les algorithmes généraux de la mise à jour et de la consultation des données. Pour la description des applications, nous donnons d'abord quelques explications, ensuite le format général, enfin les algorithmes.

5.1 Remarques préliminaires

La restructuration des données nous a obligé à apporter des modifications à la base de données descriptive. Nous définissons d'abord la structure de la B.D.D. et justifions ensuite la correspondance biunivoque entre une relation physique et un fichier. Nous prenons, enfin, en considération l'aspect temporel des données.

5.1.1 La base de données descriptive (B.D.D.)

Le but de la base de données descriptive est de mettre en oeuvre la gestion automatique de la base de données. Avant de décrire la structure conceptuelle de la B.D.D. rappelons quelques objectifs de celle-ci :

- a) décrire la structure de la base de données pour l'utilisateur, c'est à dire décrire la sémantique des relations logiques et physiques, décrire la sémantique et la syntaxe des types de données composant ces relations.
- b) décrire le contenu de la base de données, c'est à dire décrire les réalisations physiquement présentes, les versions disponibles et complètes...

- c) assurer la gestion efficace de la propriété des données, du secret, des contraintes d'intégrité en général : formules d'aggrégations, permissions d'accès en consultation ou en mise à jour...
- d) assurer la gestion efficace des applications fondamentales de la base de données; consultation, édition et mise à jour de celle-ci.
- e) assurer la gestion des applications secondaires : consultation de la base de données descriptive, en vue de permettre à l'utilisateur de connaître l'état de la base de données. La base de données descriptive contient le dictionnaire de données défini précédemment.

5.1.1.1 Description des relations de la B.D.D.

- 1°) APPLICATIONS : liste des applications que chaque utilisateur exploite grâce aux données qu'il tire de l'entrepôt.

caractéristiques :

APPLICATION (APPLIC, RELOGID, USERID, DESCRIPTION)

où APPLIC : identification de l'application

RELOGID : identification d'une relation logique utilisée dans cette application

USERID : identification de l'utilisateur

DESCRIPTION : description en clair de l'application

- 2°) FORMULE DE CALCUL : la formule concerne les données calculées au niveau du dictionnaire

caractéristiques :

COMDATA (DATAID, FORMULE, PROGR)

où DATAID : identification du type de donnée au moyen d'un nom mnémonique unique au sein d'une relation LOGIQUE

FORMULE : description de la formule selon laquelle la donnée est calculée; cette formule utilise les cinq opérations arithmétiques fondamentales (+, -, *, :, **). Cette formule peut être utilisée à des fins documentaires ou analysée et exploitée par un programme, comme un compilateur FORTRAN analyse et exploite une instruction arithmétique dans un programme.

PROGR: nom d'un programme stocké en banque des programmes et qui calcule effectivement cette donnée.

3°) COMPOSANTE D' UNE RELATION PHYSIQUE*caractéristiques :*COMPOSANT (RELPHID, DATAID, NUMDATA, TITDATA, MESURE, POSSAI, DEPLACEMENT)

où RELPHID : identification d' une relation physique dont ce type de données est une composante

DATAID : cfr. supra

NUMDATA : numéro du type de données dans le dictionnaire

TITDATA : titre ou libellé en clair de cette donnée

MESURE : indication de l' unité de mesure dans laquelle s' exprime la donnée

POSSAI : possibilité de saisie : permet d' indiquer si la donnée est saisie et possède réellement des valeurs dans la base de données. Les valeurs de POSSAI sont 'saisie', 'partiellement saisie' ou 'non saisie'.

(*) DEPLACEMENT : déplacement de la composante par rapport au début de la relation physique

4°) TYPE DE DONNEES : expression d' une propriété d' une unité ou d' une association d' unités.*caractéristiques :*DATATYPE (DATAID, DOMID, PRIMAIRE)

où DATAID : identification du type de donnée

DOMID : identification du domaine auquel appartient ce type de donnée

(**) PRIMAIRE : indique si la donnée est primaire ('p'), calculée ('c'), primaire et calculée ('d') ou ni primaire ni calculée ('r').

5°) CLE D' UNE RELATION LOGIQUE : KEYUNIT permet de faire le lien entre une relation logique et son unité-clé, que cette unité clé soit simple ou associative.*caractéristiques :*KEYUNIT (KEYID, UNITID 1, UNITID 2, CARDINAL)

où KEYID : identification du type de clé

UNITID 1 : premier type d' unités formant la clé

(*) composante ajoutée à la relation proposée par (BDER p. 210) cfr. 5.1.1.2

(**) composante dont les valeurs possibles proposées par (BDER p. 209) ont été complétées

UNITID 2 : second type d' unités formant la clé, pas nécessairement distinct de UNITID 1. Si la clé est simple, UNITID 2 contient une valeur conventionnellement neutre.

6°) PERMISSION : un utilisateur a-t-il le droit de consulter une donnée au niveau individuel ?

caractéristiques :

PERMISSION (USERID, RELPHID, AGREG)

où USERID : cfr. supra

RELPHID : cfr. supra

AGREG : indique le niveau d' agrégation auquel on doit fournir les données si l' utilisateur ne fournit pas le mot de passe
AGREG est le nombre minimum d' occurrences à cumuler. Si l' utilisateur peut 'voir' les données individuelles AGREG = '1'

7°) RELATION LOGIQUE : entité de base de consultation de l' entrepôt

caractéristiques :

RELLOG (RELOGID, TITLOG, DIMREF, SITU, NBMXVR, KEYID, READPASS)

où RELOGID : identification de la relation logique

TITLOG : titre ou libellé en clair de cette relation

DIMREF : dimension de référence : dimension temporelle des données, commune de toutes les données de cette relation

(*) SITU : (MVT) : critère de 'mouvement'. Ce critère indique si les données sont de type 'événement', 'situation' ou 'existant'.

NBMXVR : NomBre MaXimum de VeRsions conservées par le système

KEYID : cfr. supra

(**) READPASS : mot de passe en lecture. Si les données sont couvertes par le secret, l' utilisateur doit fournir le mot de passe afin de pouvoir lire les données de la relation au niveau individuel.

8°) RELATION PHYSIQUE : entité de base de la mise à jour

caractéristiques :

RELPHY (RELPHID, TITPHY, ADRPHY, USERID, FCOL, WRPASS, RELOGID, DIVULG, READPASS, NOYAU, CALCULEE)

*) la composante SITU est remplacée par MVT ce qui correspond mieux à ce qu' elle représente

**) la composante READPASS a été ajoutée cfr. 5.1.1.2

où RELPHID : identification d' une relation physique

TITPHY : titre ou nom en clair de ctte relation

ADRPHY : adresse physique de la relation.

USERID : identificateur de l' utilisateur responsable de la mise à jour de cette relation physique

FCOL : fréquence de collecte et de mise à jour des données de cette rel.

WRPASS : 'write password' : mot de passe à fournir par l' utilisateur qui désire appoter des données nouvelles à cette relation

RELOGID : identifiant de la relation logique dont est issue la relation physique

DIVULG : valeur binaire indiquant si la relation contient des données divulguables ou non

(*) READPASS : cfr. supra

(**) NOYAU : valeur binaire indiquant si la relation appartient au noyau

(***) CALCULEE : valeur binaire indiquant si la relation ne contient que des données calculées ou non.

9°) CARACTERISTIQUES STANDARDS DE DOMAINES

caractéristiques :

STAND (DOMID, TYPE, FORMINT, FORMEXT)

où DOMID : nom de domaine

TYPE : type de ce domaine (numerique, alphanum.)

FORMINT : format interne de stockage des données de ce domaine

FORMEXT : format externe de représentation des données de ce domaine

10°) TYPE D' UNITES : les populations décrites dans l' entrepôt

caractéristiques :

UNITTYPE (UNITID, CARDINAL, DATAID)

où UNITID : identificateur d' un type d' unité

CARDINAL : cardinalité de ce type d' unité dans le système
elle peut être éventuellement indéfinie

DATAID : type de données figurant comme composante primaire dans les relations où ce type d' unités est ou fait partie de l' entité-clé.

(*) composante supprimée pour des raisons expliquées en 5.1.1.2

(**) composante supprimée devenue inutile car déjà définie par critère de mouvement

(***) composante ajoutée pour des raisons expliquées en 5.1.1.2 'existant'

11°) UTILISATEUR : client-consultant et/ou fournisseur-metteur à jour de l' entrepôt

caractéristiques :

UTILISATEUR (USERID, PASSWORD, NOM)

où USERID : identificateur de l' utilisateur

PASSWORD : mot de passe donnant accès au système (pas nécessairement accès aux relations)

NOM : nom ou désignation en clair de l' utilisateur (sert pour l' édition de statistiques d' utilisation du système, de protocoles...

12°) VERSIONS PRESENTES : VERSIONS permet au système de gérer la localisation temporelle des données stockées dans l' entr.

caractéristiques :

VERSIONS (RELPHID, PREF, ETAT, CARDIN)

où RELPHID : cfr. supra

PREF : période de référence

ETAT : état de cette version complète, incomplète ou nulle

CARDIN : nombre d' occurrences effectivement présentes

Récapitons les relation utilisée :

APPLICATION (APPLIC, RELOGID, USERID, DESCRIPTION)

COMDATA (DATAID, FORMULE, PROGR)

COMPOSANT (RELPHID, DATAID, NUMDATA, TITDATA, MESURE, POSSAI, DEPLACEMENT)

DATATYPE (DATAID, DOMID, PRIMAIRE)

KEYUNIT (KEYID, UNITID 1, UNITID 2, CARDINAL)

PERMISSION (USERID, RELPHID, AGREG)

RELLOG (RELOGID, TITLOG, DIMREF, MVT, NBMXVR, KEYID, READPASS)

RELPHY (RELPHID, TITPHY, ADRPHY, USERID, FCOL, WRPASS, RELOGID, DIVULG, CALCULEE)

STAND (DOMID, TYPE, FORMINT, FORMEXT)

UNITTYPE (UNITID, CARDINAL, DATAID)

UTILISATEUR (USERID, PASSWORD, NOM)

VERSIONS (RELPHID, PREF, ETAT, CARDIN)

5.1.1.2 Justifications des cinq modifications apportées à la B.D.D

Après avoir présenté la base de données descriptive proposée par (BDER) ainsi que les modifications que nous y avons apportées, justifions l'utilité de celles-ci.

- a) composante CALCULEE de RELPHY : étant donné que le critère de calcul est devenu un critère de scission de relations logiques en relations physiques (cfr. 3.3) afin d'interdire la mise à jour directe par l'utilisateur de données calculées, nous avons ajouté la composante CALCULEE à la relation RELPHY de la B.D.D.. Cette composante prendra une valeur binaire indiquant si elle ne contient que des données calculées ou non.
- b) composante SITU de RELLOG et composante NOYAU de RELPHY : le critère de mouvement (symbolisé par la composante SITU de RELLOG) étant utilisé lors du passage du niveau infologique au niveau conceptuel (cfr. 1.2) sera remplacé par la symbolisation plus naturelle 'MVT' : rappelons qu'il peut prendre trois valeurs de mouvement c'est à dire 'événement' (stable dans le temps), 'situation' (stable durant une période de temps) et 'existant' (très peu soumis à des modifications).

Toutes les données de type 'existant' étant reprises dans le noyau, la composante NOYAU de RELPHY n'a plus de raison d'être.
- c) composante PRIMAIRE de DATATYPE : nous proposons d'ajouter les valeurs 'd' pour des données calculées et primaires, et 'r' pour des données ni calculées ni primaires à la composante PRIMAIRE de DATATYPE afin de pouvoir énumérer toutes les valeurs possibles de la composante : ceci permettra à l'utilisateur, lors de la consultation de cette relation, d'être mieux informé. De plus, l'absence de la valeur 'd' rendait impossible le traitement des données à la fois primaires et calculées.
- d) composante READPASS de RELPHY : nous avons supprimé la composante READPASS de RELPHY pour la mettre dans la relation RELLOG car l'utilisateur consultant la base de données ne connaît que les relations logiques (unités de consultation). Sans cette modification, l'utilisateur consultant aurait dû fournir le nom des relations physiques constituant la relation logique désirée ainsi que les READPASS correspondant (alors qu'il est sensé ignorer la notion de relation physique).

- e) composante DEPLACEMENT de RELPHY : nous avons ajouté la composante
DEPLACEMENT à la relation RELPHY

car il doit être possible de repérer la position exacte d' une composante dans une relation physique. Cette information ne peut pas être recouverte par la composante FORMINT (format interne) de STAND car toutes les composantes appartenant au même DOMID devraient se trouver au même déplacement par rapport au début de leur relation physique.

5.1.1.3 Utilité des relations de la base de données descriptive

Afin de montrer l' utilité des relations de la B.D.D. nous nous limiterons à donner un exemple. Celle-ci n' apparaîtra au lecteur qu' au cours des chapitres suivants.

Les relations UTILISATEUR et APPLICATION servent à reconnaître l' utilisateur. La commande de reconnaissance de l' utilisateur aura les paramètres suivants :

userid, password, applic

L' utilisateur doit être connu comme un utilisateur autorisé du système et le mot de passe doit correspondre. De plus l' application doit aussi être connue du système et correspondre à cet utilisateur.

formellement (1)

$(userid, password) \in (USERID, PASSWORD) \text{ UTILISATEUR}$

$\wedge (applic, userid) \in (APPLIC, USERID) \text{ APPLICATION}$

-
- (1) En ce qui concerne l' expression formelle des contraintes d' intégrité, nous utilisons une notation assez simple qui ne fait appel, en plus des symboles classiques utilisés en théorie des ensembles, qu' à la projection d' une relation, que nous exprimons comme suit :

(COMPOSANTE 1, COMPOSANTE 2, ...) RELATION

Toutes les contraintes d' intégrité peuvent s' exprimer par l' appartenance d' un n-uple à une relation ou à la projection sur n composantes d' une relation. Cette notation est conforme à celle proposée dans (BDER)

Les caractères minuscules' correspondent à des valeurs de composantes, les caractères MAJUSCULES correspondent à des noms de COMPOSANTES ou de RELATIONS.

Les caractères 'abdefg' correspondront dans la suite à des commentaires.

Les caractères "entre guillemets" correspondront à m' impression d' un message d' erreur

5.1.2 Correspondance biunivoque entre une relation physique et un fichier

Les relations physiques ont été implémentées sous forme de fichiers séquentiels afin de faciliter l'opération de mise à jour qui consiste en l'apport d'un certain nombre d'enregistrements (en général une version) en fin de fichier. Ainsi, un fichier séquentiel possèdera à tout moment un contenu homogène du point de vue de sa source et de sa fréquence de collecte. Dans le cas contraire, c'est à dire si nous avons établi une liaison biunivoque entre la relation logique et un fichier séquentiel, il n'aurait pas suffi d'ajouter des enregistrements en fin de fichier, en effet, celui-ci n'aurait plus été homogène car il n'aurait pas eu un nombre d'occurrences identique pour toutes les composantes de la relation correspondante. Par contre, il aurait pu servir tel quel à la consultation. Etant donné le choix d'implémenter la relation physique sous forme d'un fichier séquentiel, il nous faut lors de la consultation fusionner les fichiers contenant les données d'une relation logique (unité de consultation).

Après avoir justifié le choix de prendre la relation physique comme unité physique, expliquons celui de l'organisation séquentielle de ces unités physiques :

D'une part le prix de l'implémentation d'accès plus sophistiqués sera plus élevé, d'autre part, on ne pourra envisager comme support de stockage de l'entrepôt que des bandes magnétiques, supports séquentiels par excellence, privilégiés pour de grandes masses de données. De plus, l'optique de (BDER) a comme conséquence de pénaliser les utilisateurs qui font des 'applications' sélectives.

On pourra néanmoins accéder directement, par l'intermédiaire de la clé primaire, au noyau et à la base de données descriptive. Le noyau, outre des données signalétiques, peut éventuellement contenir d'autres types de données, par exemple pour une consultation sélective trop fréquente.

5.1.3 Incorporation de la notion de version aux relations

Bien que la notion de version appartienne à la base de données descriptive, elle est absente de la base de données dans (BDER). Pour remédier à cette absence, nous avons décidé, de façon à pouvoir différencier les versions d' une relation, d' ajouter une ou plusieurs composantes (décennie, année, mois, trimestre, jour) aux composantes constituant la clé primaire. Elles constitueront les composantes temporelles de la clé primaire. Ceci permettra, de plus, aux utilisateurs de disposer, lors de la consultation, de la valeur de la relation correspondant aux données.

La notion de version étant absente des données signalétiques (caractéristique de mouvement 'existant') celle-ci ne sera définie que pour les autres données.

5.1.3.1 Données de type 'événement'

Pour les données dont la caractéristique de mouvement est 'événement' la notion de version est modifiée et se trouve sous la forme AAMMJJ où AA est la composante année, MM est la composante mois et JJ est la composante jour. Cette date d' observation servira de critère lors de l' archivage des données les plus anciennes. Prenons l' exemple de la relation *capital passif étranger* n° 13 dans (BDER p. 188) dont la description est la suivante :

clé primaire : identifiant de l' entreprise, identifiant des autres emprunts obligataires

422037 : Montant
422038 : Taux d' intérêt
422039 : Terme
422040 : Date d' émission
422041 : Organisme créancier

La nouvelle relation (notion de version incluse) aura la description suivante : (cfr. annexe 2 rel N° 13)

clé : identifiant de l' entreprise, identifiant des autres emprunts obligataires, date d' observation (date d' émission)

422037 : Montant
422038 : Taux d' intérêt
422039 : Terme
422041 : Organisme créancier

On peut se poser la question suivante : la date d' émission fait-elle partie de la clé primaire ? Si la réponse est 'non', cela implique qu' un emprunt, par exemple, ne peut être subdivisé en plusieurs parties (avec des dates différentes)

5.1.3.2 Données de type 'situation'

Les données ayant comme caractéristique de mouvement 'situation' ont différentes dimensions de références selon le cas : décennie, année, année-trimestre, année-mois. Prenons l' exemple de la relation intitulée *dettes* (à l' ONSS) et dont le numéro est 3.1 dans (BDER) a la description suivante :

clé : identifiant de l' entreprise

440001 : nombre de mois de retard de paiements vis-à-vis de l' ONSS

440002 : montant des paiements en retard vis-à-vis de l' ONSS

Après avoir ajouté la notion de version sa description est la suivante :

clé : identifiant de l' entreprise, année-trimestre

440001 : nombre de mois de retard de paiements vis-à-vis de l' ONSS

440002 : montant des paiements en retard vis-à-vis de l' ONSS

(en supposant que sa dimension de référence soit TRIMESTRIELLE)

Pour les autres relations nous renvoyons à l' annexe 2.

Remarque

La solution présentée n' est pas la seule possible : on aurait pu faire correspondre à chaque version d' une relation un fichier, ce qui serait devenu encombrant, car le nombre de fichiers définis et utilisés dans ce cas serait beaucoup plus élevé. Cette solution implique l' indication de l' adresse physique de chaque version non plus dans la relation RELPHY de la base de données descriptive mais dans la relation VERSIONS de celle-ci.

5.2 Création de la base de données

Nous allons tout d'abord énoncer le principe général de la création de la base de données puis donner un exemple de création d'une relation.

5.2.1 Principe général

Pour créer physiquement la base de données, il faudra auparavant disposer des données décrites par le dictionnaire de données. Il s'agit tout d'abord de rassembler les données auprès des différents organismes-sources.

Deux possibilités s'offrent : la première consiste à reprendre des données de fichiers préexistants au moyen d'un interface de reformatage (1), la seconde consiste à copier dans la base de données des données déjà formatées correctement par l'organisme-source (2).

Notons que la seconde solution respecte infiniment plus l'optique proposée par (BDER) c'est à dire assurer un outil-interface avec un minimum de transformations des données.



- (1) [UTILISATEUR] [GESTIONNAIRE DE LA BASE DE DONNEES]
- (2) [UTILISATEUR] [GESTIONNAIRE B. D.]

- figure 3 -

5.2.2 Description d'un exemple de reformatage

Le cas d'un reformatage, d'un fichier fourni par l'organisme-source, par le gestionnaire de la base de données est opposé à la politique menée par (BDER). Cette occasion pouvant malgré tout se présenter, nous proposons un exemple de création des valeurs d'une relation par l'intermédiaire d'un interface de reformatage. L'interface de reformatage prend les données né-

nécessaires sur le fichier fourni par l' organisme-source et les reporte sur le fichier à créer (relation physique). Il connaît évidemment la description du fichier à créer ce qui lui permet, de plus, d' effectuer des contrôles quant au format et au type des données.

Le *fichier du registre central du commerce* dont la description se trouve à la page suivante, peut, par exemple, nous servir à construire la relation intitulée *identifiant de l' établissement* n° 1.1 (cfr. annexe 1) dont la description est la suivante :

clé : identifiant de l' établissement
 110001 : identifiant de ET
 110002 : adresse de ET
 110003 : date de création de ET, date de mise en activité
 110004 : identifiant EP dont dépend ET
 12000N : secteur d' activité N de ET

Le programme interface possèdera les valeurs de toutes les composantes de la relation c' est à dire :

NUMERO (1)

10,1 : la clé : identifiant de l' établissement
 2,3,4,5,6 : adresse de l' établissement
 7,8 : date de création (date de début)
 1 : identifiant EP dont dépend ET
 9 : secteur d' activité i de ET

Dans l' exemple qui précède nous montrons que l' organisme-source n' a qu' un seul fichier à fournir pour constituer une relation physique. Il est cependant concevable que, selon l' organisation interne de l' organisme-source, les valeurs servant à créer la relation physique soient récoltées par le programme d' interface sur base de plusieurs fichiers.

(1) les numéros correspondent à la colonne NUMERO de la page suivante

FICHER : REGISTRE CENTRAL DE COMMERCE - ETABLISSEMENTS

NU MERO	NIV	DEB	FIN	LONG	TYP	FORMAT	LGR	DEFINITION	DOC.ORIG.	ZN ORIG.	COM. - TABLES
1	1	1	2	2	N	ZD	2	Article établissement			TC 9
	2	3	9	7	N	ZD	7	Code article			
	2	10	15	6	N	ZD	6	Registre central de commerce			
	2	16	17	2	N	ZD	2	Registre local de commerce			COM 2
	2							Ressort (Greffes)			
	2							Date			
7	3	18	18	1	N		1	Date de début			
	4	19	20	2	N		2	Mois (perfo hors texte)			
	4					ZD		Année			
	3	21	21	1	N			Date de fin			
	4	22	23	2	N		2	Mois (perfo hors texte)			
	4					ZD		Année			
9	3	24	28	5	N	ZD	5	Activité			COM 7
	2	49	51	3	N	ZD	3	Activité _i (i = 1 → 5)			
	2	52	52	1	A	C	1	N° Rue			
	2	53	71	19	A	C	19	Bis			
	2	72	76	5	N	ZD	5	Rue			
	2	77	79	3	N	ZD	3	Code commune INS			
10	2	80	80	1	N	ZD	1	Pays			TS 4
	2							Type établissement			COM 3
	2							Date			TC 14
	3	81	82	2	N	ZD	2	Mois			
	3	83	84	2	N	ZD	2	Année			
	2	85	91	6	N	ZD	6	N° registre de commerce			
10	4	93	96	4	N	ZD	4	Code carte			
	2	97	100	4	N	ZD	4	N° établissement			
	2	101	104	4	N	ZD	4	Nombre d'activités			
	2	105	120	16	N	ZD	16	Réserve - code des erreurs			
	2	121	160	40	A	C	40	Nom de la commune			
	2										

5.3 Reconnaissance de l' utilisateur

Pour avoir accès à la base de données pour la consulter ou la mettre à jour, l' utilisateur doit être reconnu du système comme un utilisateur autorisé, le mot de passe et l' application doivent lui correspondre.

Cette commande présentée par (BDER p. 215) aura la forme suivante :

SIGNON userid,password,application

où SIGNON : mot clé désignant la commande

userid : identificateur de l' utilisateur

password : mot de passe propre à cet utilisateur

application : identificateur de l' application que l' utilisateur désire exploiter avec les données qu' il tire de la base de données.

Notons que si application = \emptyset il ne pourra effectuer que la mise à jour de données.

Nous proposons un algorithme pour la reconnaissance de l' utilisateur qui tient compte des contraintes d' intégrité définies dans (BDER p. 215) et utilise la base de données descriptive telle qu' elle a été modifiée en 5.1.1

Algorithme de reconnaissance de l' utilisateur

- 1) *contrôle syntaxique des paramètres de la commande*
si le contrôle syntaxique est correct aller en 2
sinon "recommencez la commande" stop
- 2) *identification de l' utilisateur*
grâce à 'userid' accès à la relation UTILISATEUR
si (userid) \in (USERID) UTILISATEUR aller en 3
sinon "utilisateur inconnu" stop
- 3) *identification du password de l' utilisateur*
si (userid,utilisateur) \in (USERID,PASSWORD) UTILISATEUR aller en 4
sinon "password incorrect recommencez" stop
- 4) *l' utilisateur a-t-il défini son application ?*
grâce à 'applic' accès à la relation APPLICATION
si (applic) = \emptyset aller en 5
si (applic,userid) \in (APPLIC,USERID) APPLICATION aller en 5
sinon "application incorrecte recommencez" stop
- 5) *la reconnaissance de l' utilisateur a été correctement effectuée*
mémoriser applic

5.4 Application de consultation

L'opération de consultation a pour entité de base la relation logique caractérisée par le fait que ses composantes ont la même unité-clé, couvrent des périodes de même durée et sont susceptibles d'être demandées en même temps.

Lorsque la relation logique est demandée telle qu'elle existe dans la base de données, elle constitue une unité de consultation MASSIVE. Ceci est, évidemment, la forme la plus simple de consultation. Il est néanmoins possible que la totalité des informations contenues dans une relation logique n'intéresse pas l'utilisateur. Dans ce cas, le système doit lui donner la possibilité de ne choisir que les informations qui l'intéressent, en d'autres termes, de faire des consultations SELECTIVES.

Rappelons que cette application permet d'effectuer :

- la consultation de l'entrepôt
- la consultation du noyau
- la consultation de la base de données descriptive.

La consultation de la base de données est composée de la consultation de l'entrepôt et de la consultation du noyau. La seule différence est que la première se fait en accès séquentiel tandis que la seconde se fait en accès direct.

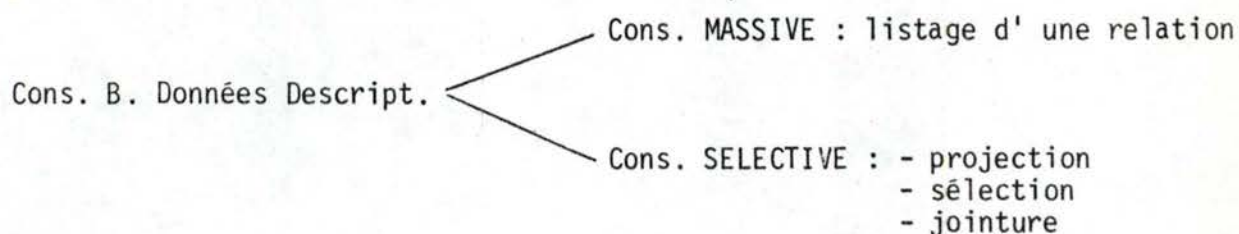
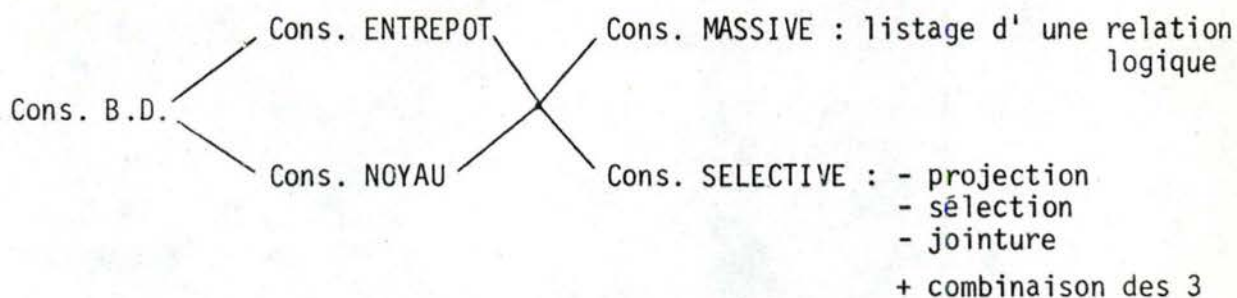
La consultation massive de la base de données est un listage d'une relation logique. Elle est à conseiller vivement car l'utilisation des autres opérations (projection, sélection, jointure, ...) s'avère plus coûteuse. C'est à l'utilisateur de choisir le prix qu'il désire payer pour consulter la base de données.

La consultation sélective de la base de données est une combinaison de commandes d'interrogation de bases de données telles que :

- projection d'une relation sur une liste de composantes
- sélection sur filtre
- jointure de deux relations sur une liste de composantes

La consultation de la base de données descriptive doit, quant à elle, présenter la description des relations et des utilisateurs de la base de données économiques régionale. Les opérations de consultation seront les mêmes que pour celle de l'entrepôt.

Schéma regroupant les opérations possibles lors de la consultation



- figure 4 -

5.4.1 Consultation de la base de données

La consultation de la base de données se présente soit, sous la forme d' une consultation massive, soit sous celle de la consultation sélective. Malgré son coût plus élevé, cette dernière peut se justifier pour deux raisons : d' une part, il existe des types de données dans une relation logique qui n' intéresse pas l' utilisateur, d' autre part celui-ci peut souhaiter disposer de types de données appartenant à des relations logiques différentes. Afin de pouvoir réaliser la consultation, nous allons examiner trois commandes de base et aborder par après d' autres opérations qui pourraient s' avérer intéressantes.

Durant l' exécution d' une combinaison des commandes de base (projection, sélection, jointure) il y aura des modifications provisoires de structure qui affecteront la base de données. Lors de l' édition du résultat, ces descriptifs seront détruits et la base de données retrouvera son état initial.

Afin de faciliter la gestion, nous réécrivons l' instruction de consultation, après chaque évaluation d' une commande de base, sous forme interne dans laquelle les relations logiques provisoires sont appelées RLOGn (n indique le numéro de la relation logique provisoire) et les relations physiques provisoires RPHYn.

L' utilisateur doit donner le mot de passe en lecture pour chaque relation logique afin de pouvoir lire les données au niveau individuel. Il faut donc vérifier pour chaque opération de consultation si les données d' une relation logique sont divulguables ou non. Ces contrôles sont effectués grâce aux données contenues dans les relations RELLOG et RELPHY de la base de données descriptive.

5.4.1.1 Listage d' une relation logique

En consultant une relation logique, l' utilisateur obtiendra toutes ses versions présentes dans la base de données. Cette opération est réalisée en trois étapes : - demande de consultation
- programme d' interface
- utilisation des résultats.

Détaillons chacune d' entre elles :

1) Demande de consultation :

Format général : LIST R

où LIST : mot clé désignant l' opération

R: nom d' une relation logique

Afin de minimiser le nombre de rapports à éditer périodiquement, l' extraction des données de la base de données consiste, en général, en une gestion par exception : la consultation ne se fera que sur demande explicite de l' utilisateur. L' initiative lui est laissée, ce qui garantit l' existence d' un besoin réel de données de sa part.

2) Programme d' interface

Ce programme va tout d' abord lire les données d' une relation logique, il les recopie ensuite sur un support selon un format externe : composante FORMEXT de la relation STAND de la B.D.D.

Un accord éventuel peut intervenir avec l' ensemble des utilisateurs sur un format externe unique de chaque relation. Les informations concernant la relation logique à éditer se trouvent dans différentes relations de la base de données descriptive telles que :

RELLOG (RELOGID,DIMREF,KEYID)
 RELPHY (RELPHID,ADRPHY,RELOGID)
 COMPOSANT (RELPHID,DATAID,DEPLACEMENT)
 KEYUNIT (KEYID,UNITID 1,UNITID 2)
 DATATYPE (DATAID,DOMID)
 UNITTYPE (UNITID,DATAID)
 STAND (DOMID,TYPE,FORMINT,FORMEXT)

Si la relation logique est composée d'une seule relation physique, il suffit alors d'éditer les données selon le format externe. Dans le cas contraire nous proposons un algorithme permettant de regrouper les données appartenant à différentes relations physiques.

Le programme interface procède alors en trois étapes :

- création d'une table de description des composantes à éditer
- tri de chaque relation physique
- fusion des relations physiques triées.

a) création d'une table de description des composantes à éditer

=====

- recherche de toutes les relations physiques de la relation logique dans RELPHY
- recherche dans COMPOSANT de toutes les composantes des relations physiques et insertion de celles-ci dans la table ainsi que de leur déplacement.
- ajouter dans la table les DATAID correspondant à UNITID 1, UNITID 2 de KEYUNIT en passant par UNITTYPE
- recherche dans STAND en passant par DATATYPE le format interne et externe des composantes de la table

La table contiendra pour chaque composante de la relation à éditer les renseignements suivants :

TABLE (DATAID, DEPLACEMENT, FORMINT, FORMEXT)

b) Tri de chaque relation physique

=====

- prendre la longueur de UNITID 1 et UNITID 2 et la transférer dans LENGTHKEY1
- prendre la longueur des composantes temporelles de la clé primaire et la transférer dans LENGHTKEY2
- trier la relation physique (se trouvant à l'adresse physique ADRPHY) de la manière suivante :

Trier la relation physique avec comme argument primaire les composantes temporelles de la clé primaire (déplacement = LENGHTKEY1, longueur = LENGTHKEY2) et comme argument secondaire optionnel les composantes non temporelles de la clé primaire (déplacement = 0, longueur = LENGTHKEY1). Ce tri présente une vue synchronique des données (argument primaire). Ce qui importe c'est que le consultant reçoive les tuples regroupés par versions, l'argument de tri secondaire importe peu.

Considérons, par exemple, les données de type 210002 : *par ET, valeur ajoutée brute au prix du marché*. Nous trouvons ce type de données dans une relation logique dont la clé est *identifiant de l'établissement*. Supposons qu'il y ait 2 établissements (ET1, ET2) pour cette relation, qui comprend en outre 3 versions. L'entrepôt comprend les valeurs suivantes :

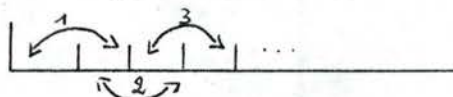
ET1	1000000	version 1
ET2	2500000	
ET1	1200000	version 2
ET2	3000000	
ET1	1500000	version 3
ET2	3250000	

Cet ordre constitue une vue synchronique des données.

Nous proposons de trier les relations physiques au moyen du tri *PAR ECHANGE* que nous allons détailler ci-dessous :

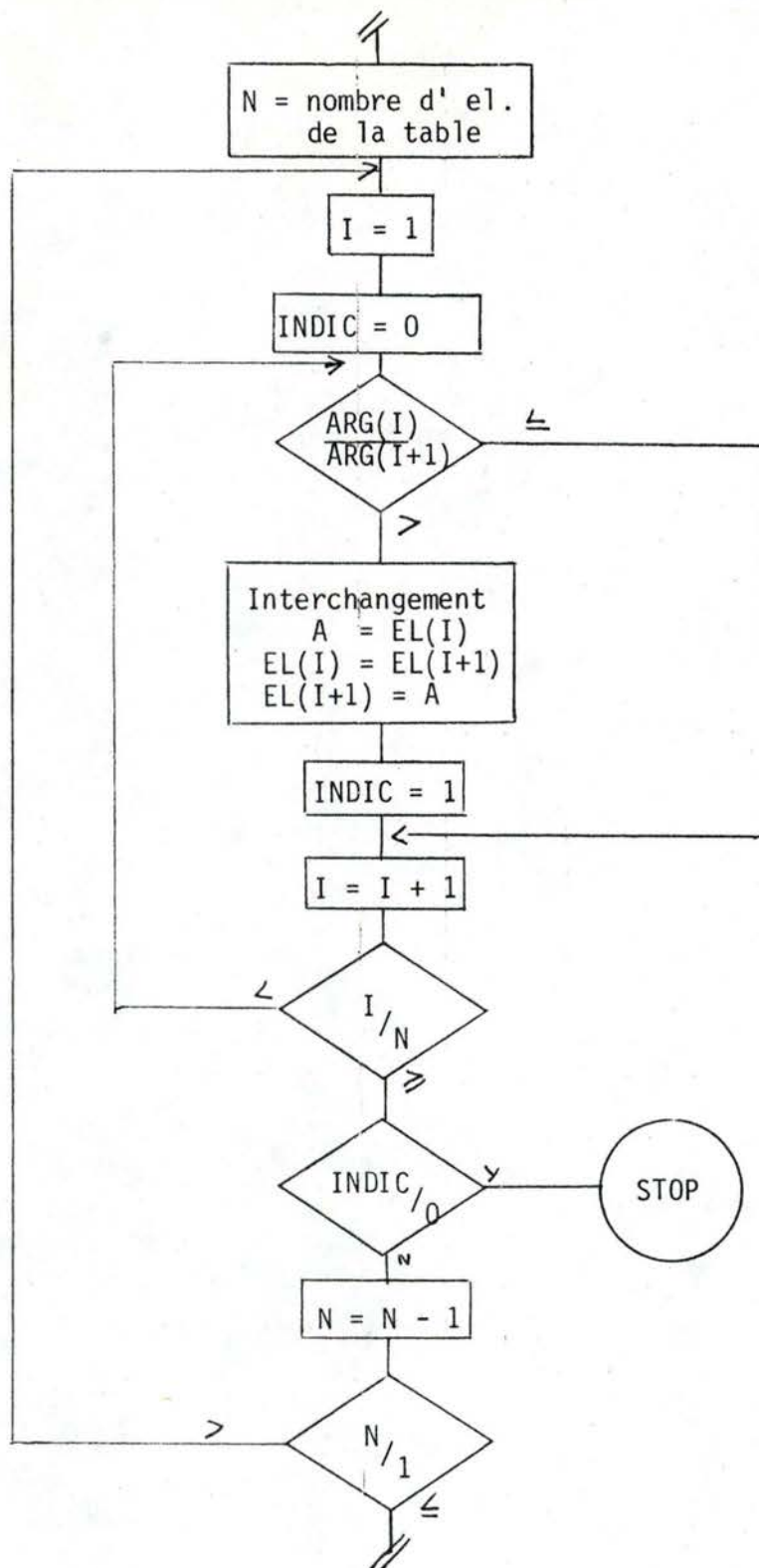
Ce tri 'par échange' est très intéressant car les fichiers sont, en principe, triés sur leurs composantes temporelles. Il permet, en effet, de gagner du temps par rapport aux méthodes de tri classique car il s'arrête après le premier passage sans interchangeement. Prenons l'exemple du tri d'une table de *n* éléments en ordre croissant :

- ° le tri par échange balaie la table par couple et progresse chaque fois d'un élément



- ° après le premier passage, on aura le plus grand des *n* éléments en dernière position
- ° on recommence en prenant les (*n*-1) premiers éléments et ainsi de suite
- ° si au cours d'un passage, il n'y a pas eu d'interchangeement, l'indicateur correspondant *n* est pas positionné et le tri s'arrête car la table est triée.

Ordinogramme du tri par échange



n = nombre d' éléments de la table à trier en ordre croissant

$INDIC$ = indicateur d' interchangement

ARG = argument de tri

$EL(I)$ = i ème élément de la table

Pour une vue diachronique des données il suffit d' inverser les arguments de tri c' est à dire trier avec comme argument primaire les composantes non temporelles de la clé et comme argument secondaire optionnel les composantes temporelles de la clé . Ainsi en reprenant l' exemple précédent nous aurions le résultat suivant :

version 1	1000000	
version 2	1200000	ET1
version 3	1500000	

version 1	2500000	
version 2	3000000	ET2
version 3	3250000	

Remarque : à la limite on peut prévoir l' inclusion dans la commande LIST un argument indiquant si l' utilisateur désire avoir une vue synchrone ou diachronique des données.

3) Utilisation des résultats

Rappelons qu' avant de pouvoir consulter une relation logique, l' utilisateur doit indiquer le nom de l' application qui exploitera les données qu' il tire de l' entrepôt, afin d' assurer un certain contrôle. Formellement : $(applic, relation) \in (APPLIC, RELOGID) APPLICATION$

5.4.1.2 Consultation sélective

La consultation sélective permet une combinaison des trois commandes de base suivantes :

- projection d' une relation sur une liste de composantes
- sélection sur filtre
- jointure de deux relations sur une liste de composantes

Pour chacune de ces commandes nous donnerons le format général, l' algorithme permettant de vérifier les contraintes d' intégrité portant sur les paramètres de la commande, les répercussions sur la base de données descriptive et l' algorithme de manipulation des données réelles.

Par la suite, nous présenterons d' autres commandes de consultation, moins utiles dans le cadre de notre application, avant d' aborder l' édition des données.

5.4.1.2.1 Projection d' une relation sur une liste de composantes

A) Format général (cfr. 3.6.1.2)

PROJ (R (L))

où PROJ : mot clé correspondant à l' opération de projection

R : nom d' une relation logique

L : liste de composantes de cette relation

Notons que si les composantes primaires sont absentes de L, elles sont ajoutées afin d' éviter le risque d' oubli.

B) Algorithme permettant de vérifier les contraintes d' intégrité portant sur la relation et la liste des composantes

L' algorithme vérifie :-si le nom de la relation logique est connue du système

-si l' application peut exploiter les données de cette relation

-si les composantes projetées sont connues du syst.

-si la liste des composantes contient toutes les composantes primaires.

B1) *le nom de la relation logique doit être connu de la B.D.D.*

si (R) \in (RELOGID) RELLOG aller en B2

sinon "R relation inconnue du système" stop

B2) *l' application (APPLIC de la commande SIGNON) peut-elle exploiter les données de cette relation logique ?*

si (applic,R) \in (APPLIC, RELOGID) APPLICATION aller en B3

sinon "applic ne peut pas exploiter les données de R" stop

B3) *la liste des composantes contient-elle toutes les composantes primaires ?*

la liste des composantes doit contenir toutes les composantes primaires afin d' éviter des risques d' oubli; une exception est tolérée, lorsque la liste des composantes est constituée seulement de composantes primaires.

Grâce à (keyid,R) \in (RELOGID, KEYID) RELLOG on accède à

KEYUNIT (KEYID, UNITID 1, UNITID 2)

si unitid 1 n' appartient pas à L, alors on l' ajoute à L sauf s' il n' y a pas de composantes non primaires dans L

si unitid 2 n' appartient pas à L, alors on l' ajoute à L sauf s' il n' y a pas de composantes non primaires dans L

si dimref (composantes temporelles) n' appartient pas à L, alors on l' ajoute à L sauf s' il n' y a pas de composantes non primaires dans L.

B4) *recherche de toutes les relations physiques de R*

balayer RELPHY afin de trouver toutes les relations physiques de R
mettre dans I les relphid pour lesquelles

$(\text{relphid}, R) \in (\text{RELPHID}, \text{RELOGID}) \text{ RELPHY}$

B5) *recherche de toutes les composantes des relations physiques de I*

balayer COMPOSANT afin de trouver toutes les composantes (DATAID) des
relations de I et les mettre dans C.

$\forall \text{relphid} \in I, (\text{relphid}, \text{dataid}) \in (\text{RELPHID}, \text{DATAID}) \text{ COMPOSANT}$

B6) *vérifier si tous les éléments de L appartiennent à C*

si les éléments de L \in à C aller en C1

sinon "composante ...absente de la relation R" stop

C) Description en base de données descriptive

Il s'agit de créer - une relation logique provisoire

- une relation physique provisoire

- les composantes de la relation physique provisoire.

C1) *insertion dans RELLOG d'un tuple décrivant la relation logique provisoire.*

le tuple inséré aura la forme (rlogi, keyid) RELLOG

si keyid de R \in L, keyid de R = keyid de rlogi

si keyid de R \notin L et si unitid 1 ou unitid 2 \in L

alors keyid correspondant à rlogi = unitid 1 ou unitid 2

si ni unitid 1 ni unitid 2 \in L alors

keyid corresp. à rlogi = valeur partic. neutre

et dimref corresp. à rlogi = L (1)

i = ième ajoute d'une relation logique provisoire

(1) L ne contiendra dans ce cas que des composantes temporelles de la
clé provisoire

C2) *insertion dans RELPHY d'un tuple décrivant la relation physique provisoire*

le tuple inséré aura la forme (rphysi, adrphysi, rlogi) RELPHY

i = ième ajoute d'une relation physique provisoire

adrphysi = nouvelle adresse physique à laquelle on va stocker les
données résultant de la projection

C3) insérer pour chaque élément de L un tuple dans COMPOSANT

$\forall a \in L, (\text{rphysi}, a) \text{ COMPOSANT}$

D) Algorithme de manipulation des données réelle

Si la projection est une commande simple ou la dernière commande d' une combinaison, on exécutera l' algorithme d' édition (cfr. C de 5.4.1.2.6). Dans le cas contraire, on appellera les points C1, C2, C3 de l' algorithme d' édition, avant de stocker les données selon le format interne à une adresse physique qu' on va indiquer dans (rphi, adrphy) RELPHY

5.4.1.2.2 Sélection sur filtre

A) Format général (cfr. 3.6.1.3)

SEL (R { L op V }) ;

où SEL = mot clé correspondant à l' opération de sélection

R = nom d' une relation logique

op = opérateur de sélection

L = nom d' une composante de R

V = valeur de cette composante

{L op V} = filtre

B) Algorithme permettant de vérifier les contraintes d' intégrité portant sur les paramètres de la commande

On vérifie si - le nom de la relation logique est connue

- l' application peut exploiter les données de cette relation
- toutes les composantes contenues dans le filtre sont connues du système comme appartenant à la relation logique
- les valeurs du filtre sont du bon type, du bon format

B1) *le nom de la relation logique doit être connu de la B.D.D.*

si (R) \in (RELOGID) RELLOG aller en B2

sinon "R: relation logique inconnue" stop

B2) *l' application définie dans SIGNON peut-elle exploiter les données de la relation logique ?*

si (applic, R) \in (APPLIC, RELOGID) APPLICATION aller en B3

sinon "applic ne peut pas exploiter les données de R" stop

B3) *rechercher toutes les relations physiques de R*

mettre dans I les relphid pour lesquels

(relphid,R) \in (RELPHID, RELOGID) RELPHY

B4) balayer COMPOSANT afin de trouver tous les dataid (ncm de composante) des éléments de I et les mettre dans C

relphid \in I, (relphid,dataid) \in (RELPHID, DATAID) COMPOSANT

B5) *les éléments de L appartiennent-ils tous à C ?*

si les éléments de $L \in C$ aller en B6

sinon "composante ... absente de R" stop

B6) *recherche des domaines de chaque type de données de L*

pour tous les éléments de L, balayer DATATYPE afin de trouver leur domid (domaine).

B7) *les éléments de L sont-ils de même type et de même format que leurs valeurs correspondantes ?*

vérifier à l'aide de STAND si les éléments de L sont de même type et de même format que leurs valeurs correspondantes

si oui aller en C1

sinon "erreur sur le type ou le format de la composante ...de R" stop

C) Description en base de données descriptive

Il faut créer - une relation logique provisoire

- une relation physique provisoire

- les composantes de la relation physique provisoire

C1) *insertion dans RELLOG d' un tuple décrivant la relation logique prov.*

le tuple inséré aura la forme (rlogi) RELLOG

C2) *insertion dans RELPHY d' un tuple décrivant la relation physique prov.*

le tuple inséré aura la forme (rphi, adrphy, rlogi) RELPHY

i = ième relation provisoire

C3) *insertion dans COMPOSANT d' un tuple décrivant chaque élément de C*

$\forall c \in C$, insérer (rphi, c) COMPOSANT

D) Algorithme de manipulation des données réelle

Si la sélection est une commande simple ou la fin d' une opération de consultation, on appliquera l' algorithme d' édition cfr. C de 5.4.1.2.6).

Sinon, on exécute les points C1, C2, C3 de cet algorithme et on mémorise les données à l' adresse physique indiquée dans (rphi, adrphy) RELPHY

5.4.1.2.3 Jointure de deux relation sur une liste de composantes

A) Format général (cfr. 3.6.1.4)

JOIN (R1 (L1) * R2 (L2))

où JOIN = mot clé correspondant à l'opération de jointure

R1,R2 = noms de relations logiques

L1 = liste de composantes de R1

L2 = liste de composantes de R2

B) Algorithme permettant de vérifier les contraintes d'intégrité portant sur les paramètres de la commande

On vérifie si - le nom de la relation logique est connue

- l'application définie peut exploiter les données de cette relation
- les composantes appartiennent à leur relation respective
- les composantes qui occupent le même rang dans les listes sont de même type et de même format
- l'opération de jointure ne conduit pas à créer une relation dont la clé a plus de 2 composantes primaires non temporelle

B1) *le nom de la relation logique doit être connu de la B.D.D.*

si (R) \in (RELOGID) RELLOG aller en B2

sinon "R relation log. inconnue " stop

B2) *l'application définie dans SIGNON peut-elle exploiter les données de la relation R ?*

si (applic,R) \in (APPLIC, RELOGID) APPLICATION aller en B3

sinon "application ne peut pas exploiter les données de R" stop

B3) *rechercher toutes les relations physiques de R*

balayer RELPHY afin de trouver toutes les relations physiques de relogid = R1 (et R2) et les mettre dans I1 (et I2).

$\forall a \in (R1, R2), (relphid, a) \in (RELPHID, RELOGID) RELPHY$

B4) balayer COMPOSANT afin de trouver tous les dataid (nom de composante) correspondant aux éléments de I1 (et I2) et les mettre dans C1 (et C2).

$\forall b \in (I1 \cup I2), (b, dataid) \in (RELPHID, DATAID) COMPOSANT$

B5) vérifier si tout élément de L1 (L2) appartient à C1 (C2)

si oui aller en B6

sinon "composante... absente de R1 ou R2" stop

- B6) *recherche des domaines (domid) de tous les types de données de L1, L2*
 balayer DATATYPE afin de trouver le domid des éléments de L1, L2
 $\forall c \in (L1 \cup L2), (c, \text{domid}) \in (\text{DATAID}, \text{DOMID}) \text{ DATATYPE}$
- B7) *vérifier que les dataid qui occupent le même rang dans L1, L2*
appartiennent au même domid
 si les dataid qui occupent le même rang dans les listes ont le même
 type et le même format interne aller en B8
 sinon "composante ...de R1 et comp...de R2 n' ont pas m domid" stop
- B8) prendre dans RELLOG les keyid de R1 et R2 et les mettre respectivement
 dans K1 et K2
 $(R1, K1) \in (\text{RELOGID}, \text{KEYID}) \text{ RELLOG}$
 $(R2, K2) \in (\text{RELOGID}, \text{KEYID}) \text{ RELLOG}$
- B9) prendre dans KEYUNIT les unitid correspondant à R1, R2
 pour $(K1, U11, U12) \in (\text{KEYID}, \text{UNITID } 1, \text{UNITID } 2) \text{ KEYUNIT}$
 mettre unitid 1 dans U11 et unitid 2 dans U12
 pour $(K2, U21, U22) \in (\text{KEYID}, \text{UNITID } 1, \text{UNITID } 2) \text{ KEYUNIT}$
 mettre unitid 1 dans U21 et unitid 2 dans U22
- B10) Toute opération de jointure qui conduit à créer une relation dont
 la clé compte plus de 2 types d' unités est interdite. NBCLE (ini-
 tialisé à 0) est le nombre de types d' unités (ou composantes pri-
 maires non temporelles) présentes dans la clé.
 si $U11 \in L1$ et si à U11 correspond soit U21 soit U22 dans L2
 alors $\text{NBCLE} = \text{NBCLE} + 1$
 sinon aller en B11
 si $U11 \notin L1$ alors $\text{NBCLE} = \text{NBCLE} + 1$
- B11) si $U12 \in L1$ et si U12 correspond soit à U21 soit à U22 dans L2
 alors $\text{NBCLE} = \text{NBCLE} + 1$
 sinon aller en B12
 si $U12 \notin L1$ alors $\text{NBCLE} = \text{NBCLE} + 1$
- B12) si $U21$ (ou $U22$) $\in L2$ et s' ils ne correspondent ni à U11 ou U12 dans L1
 alors $\text{NBCLE} = \text{NBCLE} + 1$ et aller en B13
- B13) si $\text{NBCLE} > 2$ (*l' opération de jointure conduit à créer une relation*
dont la clé compte plus de 2 types d' unités)
 alors "jointure trop coûteuse" stop
 sinon aller en D1

C) Description en base de données descriptive

Il faut créer - une relation logique provisoire

- une relation physique provisoire

- les composantes de la relation physique provisoire

C1) *insertion dans RELLOG d' un tuple décrivant la relation logique prov.*

insérer un tuple de la forme

(rlogi, keyid) dans (RELOGID, KEYID) RELLOG

keyid \in { U11, U12, U21, U22 }

C2) *insertion dans RELPHY d' un tuple décrivant la relation physique prov.*

insérer un tuple de la forme

(rphi, adrphi, rlogi) dans (RELPHY, ADRPHY, RELOGID) RELPHY

i = ième relation provisoire

C3) insérer dans COMPOSANT $\forall a \in (C1 \cup C2)$ un tuple de la forme

(rphi, a, déplacement) dans (RELPHID, DATAID, DEPLACEMENT) COMPOSANT

D) Algorithme de manipulation des données réelles

D1) pour les deux relations logiques R1 et R2 on appelle C1, C2, C3 de l' algorithme d' édition (cfr. 5.4.1.2.6) et stocke leurs données aux adresses physiques respectives.

Deux relations logiques et physiques provisoires seront définies pour R1 et R2 :

(rlogi) dans (RELOGID) RELLOG pour R1

(rlogi) dans (RELOGID) RELLOG pour R2

(rphi) dans (RELPHID) RELPHY pour R1

(rphi) dans (RELPHID) RELPHY pour R2

$\forall a \in R1$, (rphi, a) dans (RELPHID, DATAID) COMPOSANT

$\forall a \in R2$, (rphi, a) dans (RELPHID, DATAID) COMPOSANT

D2) Ayant regroupé toutes les données de R1 ainsi que celles de R2 et ayant défini des synonymes internes pour R1 et R2 (rlogi(R1), rlogi(R2)), nous utiliserons les tables de description des composantes de R1 et R2 obtenues dans D1.

D3) *tri des relations logiques R1 et R2*

on trie la relation R1 suivant les valeurs de L1

on trie la relation R2 suivant les valeurs de L2

D4) *fusion des données des 2 relations triées*

on fusionne les 2 relations en ne recopiant dans la relation résultat que les tuples de R1 et R2 tels que la valeur de la liste des composantes de L1 est la même que celle de L2.

- D5)-si la jointure est une commande simple ou la fin d' une opération de consultation, on édite les données sur un support selon le format externe indiqué dans les tables.
- sinon, les données sont stockées selon le format interne, avant d'être utilisées lors d' étapes ultérieures de l' exécution de l' opération de consultation.

5.4.1.2.4 Adjonction de nouvelles composantes (cfr. 3.6.1.5)

L' adjonction de nouvelles composantes consiste à ajouter provisoirement une ou plusieurs composantes de type numérique, qui sont des fonctions utilisant comme opérandes des composantes de cette relation.

Vu la philosophie générale du système de base de données qui veut limiter le nombre de transformations de données, nous ne voyons pas l' intérêt de réaliser cette opération dans le cadre de l' application de consultation. Cependant, l' utilisateur peut toujours réaliser l' adjonction de composantes sous sa propre responsabilité. Toutefois, si cette opération est fréquemment demandée par des utilisateurs, nous pouvons soit implémenter cette opération, soit ajouter réellement une composante à la relation concernée. (cfr. 5.5.3.3)

5.4.1.2.5 Autres opérations de consultation

En complément des quatre opérations de consultation définies précédemment, on pourrait réaliser d' autres opérations si la fréquence de demandes de la part des utilisateurs est assez élevée. Ce sont des opérations telles que :

- Union
- Intersection
- Différence
- Fonctions verticales (somme, moyenne, maximum, ...)

Pour plus de détails au sujet de ces fonctions nous renvoyons au chapitre 6 (Lapin) et au chapitre 7 (System R).

5.4.1.2.6 Edition

L' application de consultation comporte comme dernière étape l' édition des données. L' algorithme d' édition remplit deux rôles : le premier consiste à gérer le secret des données en procédant aux agrégations qui s' imposent, le second est d' extraire les données de la base de données en se basant sur les informations de gestion des relations à éditer.

A) Autorisation de consulter les données individuelles

Avant d' éditer les données, il s' agit de vérifier si l' utilisateur a le droit de 'voir' les données individuelles. S' il n' a pas fourni de mot de passe en lecture, la composante AGREG de la relation PERMISSION de la base de données descriptive indique le niveau d' agrégation auquel on doit fournir les données de type numérique. Les valeurs de cette composante et leur signification sont à définir lors de discussions avec les utilisateurs.

B) Données de gestion des relations à éditer

Les informations décrivant la relation à éditer se trouvent dans différentes relations de la B.D.D. :

RELLOG (RELOGID, DIMREF, KEYID)
 RELPHY (RELPHID, ADRPHY, RELOGID)
 COMPOSANT (RELPHID, DATAID, DEPLACEMENT)
 KEYUNIT (KEYID, UNITID 1, UNITID 2)
 UNITTYPE (UNITID, DATAID)
 DATATYPE (DATAID, DOMID)
 STAND (DOMID, TYPE, FORMINT, FORMEXT)

C) Algorithme d' édition

L' algorithme d' édition permet de contrôler l' édition sur un support des données consultées en se basant sur les données de gestion. Il définit d' abord une table de description des composantes et procède ensuite au tri des données avant d' éditer celles-ci.

C1) Création d' une table de description des composantes à éditer

- recherche de toutes les relations physiques de la relation logique à éditer, dans RELPHY
- recherche dans COMPOSANT toutes les composantes des relations physiques et les insérer avec leur déplacement dans la table.

- ajouter dans la table des dataid correspondant à la keyid
- recherche dans STAND, en passant par DATATYPE, des formats internes et externes des composantes.

La table contiendra, donc, des renseignements pour chaque composante de la relation à éditer : TABLE (DATAID, DEPLACEMENT, FORMINT, FORMEXT)

C2) tri de chaque relation physique

- prendre la longueur de keyid et la transférer dans LENGTHKEY1
- prendre la longueur des composantes temporelles de la clé primaire et la transférer dans LENGTHKEY2
- trier la relation physique (se trouvant à l'adresse (rphi,adrphy) ∈ (RELPHID, ADRPHY) RELPHY) avec comme argument primaire les composantes temporelles de la clé primaire (déplacement = LENGTHKEY1 et longueur dans un tuple = LENGTHKEY2) et comme argument secondaire les composantes non temporelles de la clé primaire (déplacement = 0 et longueur = LENGTHKEY1)

C3) fusion des relations physiques triées

Les 'n' relations physiques triées sur le même argument sont regroupées en une seule relation sous la forme suivante :

(unitid 1, unitid 2, composantes temporelles de la clé, composantes non primaires de la lère relation physique,
..., composantes non primaires de la nème relation physique)

C4) édition des données

Les données sont, ensuite, éditées sur un type de support spécifié par l'utilisateur selon le format externe indiqué dans STAND

C5) élimination des informations provisoires

Il ne reste plus qu'à éliminer les éléments correspondants à la création provisoire des relations logiques et physiques intermédiaires dans RELLOG, RELPHY et COMPOSANT, pour l'opération de consultation concernée.

5.4.2 Consultation de la base de données descriptive

La consultation de la base de données descriptive consiste en l' une des quatre opérations suivantes :

- listage d' une relation
- projection
- sélection
- jointure

Nous allons définir les informations décrivant les relations de la B.D.D. ainsi qu' un algorithme pour chacune des opérations de consultation et le format général de celles-ci.

5.4.2.1 Informations décrivant les relations de la B.D.D.

Les informations décrivant les relations de la base de données descriptive sont indispensables afin d' assurer un certain contrôle. Pour toute relation de la B.D.D., on stocke dans une table le nom et l' adresse physique à laquelle elle se trouve. Pour chaque composante on définit leur nom, leur format interne et externe, leur type et déplacement.

5.4.2.2 Listage d' une relation de la B.D.D.

A) Format général :

LISTBDD R

où LISTBDD : mot clé correspondant à l' opération de listage (B.D.D.)

R : nom d' une relation de la B.D.D.

B) Algorithme d' édition

On édite les valeurs des composantes de cette relation (R), dont on connaît l' adresse physique, selon le format externe sur un support.

5.4.2.3 Projection d' une relation de la B.D.D. sur une liste de composantes

A) Format général :

PROJBDD (R (L))

où PROJBDD : mot clé correspondant à l' opération de projection (B.D.D.)

R : nom d' une relation de la B.D.D.

L : liste de composantes de R

B) Algorithme d' édition

- vérifier si R est le nom d' une relation de la B.D.D.
- vérifier si les noms des composantes de L sont des noms de composantes de R
- chercher à l' adresse physique de R, les données correspondant à L, et édition de celles-ci selon le format externe sur un support.

5.4.2.4 Sélection sur filtreA) Format général :

SELBDD (R { L op V })

où SELBDD : mot clé correspondant à l' opération de sélection (B.D.D.)

R : nom d' une relation de la B.D.D.

L : nom d' une composante de R

V : expression

op : opérateur de comparaison (< ≤ = ≥ > ≠)

{ L op V } : filtre

B) Algorithme d' édition

- vérifier si R est le nom d' une relation de la B.D.D.)
- vérifier si L est le nom d' une composante de la relation R
- vérifier si V est du même type et de même longueur que la composante correspondante dans L
- chercher à l' adresse physique de R et éditer, tous les tuples qui satisfont à la condition { L op V }, selon le format externe sur un support.

5.4.2.5 Jointure de deux relationsA) Format général :

JOINBDD (R1 (L1) * R2 (L2))

où JOINBDD : mot clé correspondant à l' opération de jointure (B.D.D.)

R1, R2 : noms de relations de la B.D.D.

L1 : liste de composantes de R1

L2 : liste de composantes de R2

B) Algorithme d' édition

- vérifier si R1 et R2 sont des noms de relation de la B.D.D.
- vérifier si toutes les composantes de L1 sont définies pour R1
L2 sont définies pour R2
- vérifier si le type et le format sont les mêmes pour des composantes occupant le même rang dans L1 et L2.
- trier R1 sur les valeurs de L1
R2 sur les valeurs de L2
- fusionner les deux relations en recopiant tous les tuples tels que $L1 = L2$, sur un support selon le format externe.

5.5 Mise à jour de la base de données

La mise à jour de la base de données telle qu'elle est présentée en 3.6.2 permet quatre types d'opération :

- la mise à jour ponctuelle du noyau
- la mise à jour ponctuelle de l'entrepôt des valeurs
- la mise à jour ponctuelle de la base de données descriptive
- la mise à jour massive de l'entrepôt des valeurs

Les trois premières opérations sont exclusivement réservées au gestionnaire de la base de données afin d'assurer la cohérence de la base de données. Nous allons définir la mise à jour ponctuelle et massive, présenter pour chaque type d'opération un algorithme de résolution.

5.5.1 Mise à jour ponctuelle

La mise à jour ponctuelle consiste en l'ajoute, la modification ou la suppression de certaines données d'un ou de plusieurs tuples de relations du noyau, de l'entrepôt des valeurs et de la base de données descriptive. Elle ne pourra se faire que par l'intermédiaire du gestionnaire de la base de données.

Alors que la mise à jour massive, consiste en l'ajoute de grandes quantités de données à l'entrepôt des valeurs, la mise à jour ponctuelle autorise la modification d'un nombre limité de données.

Les trois types d'opération ponctuelle envisagés sont :

- la mise à jour du noyau (caractéristique de mouvement 'existant')
- la mise à jour de l'entrepôt des valeurs (caractéristique de mouvement 'événement' ou 'situation')
- la mise à jour de la base de données descriptive

Notons que la première partie de l'algorithme de mise à jour du noyau et de l'entrepôt des valeurs est commune. Nous présenterons tout d'abord leur tronc commun et ensuite leurs différences.

5.5.1.1 Mise à jour ponctuelle de la base de données

Les contrôles suivants sont effectués lors de la mise à jour ponctuelle de la base de données :

- contrôle syntaxique des paramètres de la commande
- vérification de l' existence de la relation à mettre à jour
- vérification du critère de calcul de la relation physique
- vérification du mot de passe de l' utilisateur
- de plus, il y a appel de la routine de mise à jour ponctuelle du noyau ou celle de mise à jour ponctuelle de l' entrepôt selon que la caractéristique de mouvement est respectivement 'existant' ou 'situation' où d' autres contrôles sont effectués.

5.5.1.1.1 Format général :

MAJ relation,writepass

où MAJ : mot clé correspondant à l' opération de mise à jour
 relation : nom de la relation physique à mettre à jour
 writepass : mot de passe en écriture de la relation

La valeur de la clé précise le type d' opération à effectuer : modification, ajoute ou suppression. Pour cette dernière, le gestionnaire de la base de données fournira, de plus, un indicateur de suppression positionné à 1.

5.5.1.1.2 Algorithme pour la mise à jour ponctuelle de la base de données

Rappelons, que grâce à la reconnaissance de l' utilisateur, nous connaissons userid et que les informations qui servent à gerer la mise à jour sont contenues dans les relations de la base de données descriptive (cfr. 5.1.1.1).

- a) *contrôle syntaxique des paramètres de la commande*
 si contrôle syntaxique ok aller en b
 sinon "erreur syntaxique, recommencez" stop
- b) *vérification de l' existence de la relation*
 grâce à relation accès à RELPHY
 si (relation) \in (RELPHID) RELPHY aller en c
 sinon "relation inconnue du système" stop
- c) *vérification du critère de calcul*
 si (relation,1) \in (RELPHID, CALCULEE) RELPHY aller en d
 sinon "relation calculée non modifiable directement" stop

- d) l' utilisateur a-t-il le droit de mettre cette relation à jour ?
 si (relation,userid) \in (RELPHID, USERID) RELPHY aller en e
 sinon "relation non modifiable par cet utilisateur" stop
- e) l' utilisateur a-t-il fourni le mot de passe correspondant à la relation ?
 si (relation,userid,writepass) \in (RELPHID,USERID,WRPASS) RELPHY aller en f
 sinon "mot de passe incorrect pour la relation" stop
- f) détermination de la caractéristique de mouvement de la relation
 grâce à relation rechercher dans RELLOG la relation logique (relogid)
 à laquelle appartient la relation physique
 si (relogid,'ex') \in (RELOGID,MVT) RELLOG aller en 5.5.1.1.3, il s' agit
 de la mise à jour d' une relation appartenant au noyau
 si (relogid,'sit' ou 'ev') \in (RELOGID,MVT) RELLOG aller en 5.5.1.1.4, il
 s' agit de la mise à jour d' une relation de l' entrepôt.

5.5.1.1.3 Algorithme pour la mise à jour ponctuelle du noyau

En plus des contrôles effectués en 5.5.1.1.1 d' autres vérifications sont opérées :

- contrôle de la syntaxe des paramètres fournis
- vérification de l' appartenance des composantes fournies à la relation à mettre à jour
- reconnaissance du type d' opération et contrôle du type et du format de chaque composante

- 1) entrez comme paramètres : le nom des composantes à mettre à jour
 entrez le caractère '#' si on désire mettre toutes les composantes à jour
- 2) vérification syntaxique des paramètres (séparation par une virgule, par un blanc par exemple
 si contrôle syntaxique ok aller en 3
 sinon "paramètres incorrects" aller en 1 (au maximum 2 fois sinon stop)
- 3) vérification de l' appartenance des composantes fournies à la relation à mettre à jour
 si le caractère '#' a été fourni aller en 4
 grâce à relphid accès à COMPOSANT
 soit L la liste des composantes fournie, $\forall c \in L$:
 si (relation,c) \in (RELPHID,DATAID) COMPOSANT aller en 4
 sinon "composante n' appartenant pas à relation" aller en 1 (au maximum 2 fois et puis stop)

- 4) *recherche du type et du format interne de chaque élément de L*
 $\forall c \in L$, recherche du domaine (domid)
 prendre $(c, \text{domid}) \in (\text{DATAID}, \text{DOMID})$ DATATYPE puis
 prendre $(\text{domid}, \text{type}, \text{formint}) \in (\text{DOMID}, \text{TYPE}, \text{FORMINT})$ STAND
- 5) *reconnaissance du type d'opération : modification, ajoute ou suppression*
 entrez la valeur de la clé, et du reste du tuple
 accès aux données réelles grâce à l'adresse physique (ADRPHY)
 si la valeur de la clé existe dans le fichier aller en 6, (il s'agit
 d'une modification ou d'une suppression)
 sinon aller en 7, (il s'agit d'une ajoute)
- 6) *modification de données dans un tuple de la relation*
 entrez l'indicateur de suppression
 si INDIC = 0 alors modification des valeurs réelles avec contrôle du type
 et du format de chaque composante à modifier, aller en 8
 si INDIC = 1 alors archiver le tuple
 supprimer le tuple et aller en 8
- 7) *ajoute d'un tuple à la relation*
 ajoute du tuple avec vérification du type et du format de chaque composante à mettre à jour
- 8) *est-ce la fin de la mise à jour ?*
 si il y a encore des tuples à mettre à jour aller en 5
 sinon FIN DE LA MISE à JOUR DU NOYAU stop

Remarque : on envisage d'archiver les tuples supprimés dans les relations du noyau afin de conserver l'historique de toutes les données du noyau.

5.5.1.1.4 Algorithme de mise à jour ponctuelle de l'entrepôt des valeurs

En plus des contrôles effectués en 5.5.1.1.1 d'autres vérifications sont opérées :

- contrôle de la syntaxe des paramètres fournis
- vérification de l'appartenance des composantes fournies à la relation à mettre à jour
- reconnaissance du type d'opération et contrôle du type et du format de chaque composante
- test de la présence obligatoire
- existence de calcul à effectuer (uniquement pour les données ayant 'situation' comme caractéristique de mouvement)

- la mise à jour de l'entrepôt consiste à mettre à jour des données ayant comme caractéristique de mouvement 'événement' ou 'situation'.
- si (relogid,'ev') \in (RELOGID,MVT) RELLOG aller en A
 si (relogid,'sit') \in (RELOGID,MVT) RELLOG aller en B

A) Algorithme pour la mise à jour de données 'événement'

Pour les points 1, 2, 3, 4 nous renvoyons en 5.5.1.1.3 points 1, 2, 3, 4.

- 1) entrez les paramètres : nom des composantes à mettre à jour
 L contient la liste des composantes à mettre à jour
- 2) vérification syntaxique des paramètres
- 3) vérification de l'appartenance des composantes fournies à la relation
 à mettre à jour
- 4) recherche du type et du format interne de chaque élément de L
- 5) reconnaissance du type d'opération : modification, ajoute ou suppression
 Entrez la valeur de la clé, et du reste du tuple
 accès aux données réelle grâce à ADRPHY (adresse physique)
 si la valeur de la clé existe dans le fichier contenant la relation
 aller en 6 (il s'agit d'une modification ou d'une suppression)
 sinon aller en 7 (il s'agit d'une ajoute)
- 6) modification de données dans un tuple de la relation
 entrez l'indicateur de suppression
 si INDIC = 0 alors modification des valeurs réelles avec contrôle du
 type et du format de chaque composante à modifier
 et aller en 10
 si INDIC = 1 alors archivage du tuple
 suppression du tuple et aller en 10
- 7) ajoute d'un tuple à la relation après vérification du TEST DE LA
 PRESENCE OBLIGATOIRE (cfr.5.5.2) qui consiste en :
 grâce à keyid trouvé par (relogid,keyid) \in (RELOGID,KEYID) RELLOG
 on accède à la relation KEYUNIT
 si (keyid,composante non temporelle 1 de la clé, composante non temp. 2) \in
 (KEYID,UNITID 1,UNITID 2) KEYUNIT aller en 8
 sinon aller en 9

- 8) *les valeurs des unitid sélectionnés en 7 doivent appartenir au noyau*
 si les valeurs des unitid sélectionnés en 7 appartiennent au noyau
 alors aller en 9
 sinon "création du tuple interdite car absence de données signalétiques"
 aller en 10
- 9) *ajoute effective du tuple à la relation physique*
 ajoute du tuple à la relation avec contrôle du type et du format
 interne de chaque composante ajoutée
- 10) *est-ce la fin de la mise à jour ponctuelles de l'entrepôt pour les données 'événement' ?*
 si il y a encore des tuples à mettre à jour aller en 5
 sinon stop

B) Algorithme pour la mise à jour de données 'situation'

Pour les points 1, 2, 3, 4 nous renvoyons en 5.5.1.1.3 points 1, 2, 3, 4

- 1) *entrez les paramètres : nom des composantes à mettre à jour*
- 2) *vérification syntaxique des paramètres*
- 3) *vérification de l'appartenance des composantes fournies à la relation à mettre à jour*
- 4) *recherche du type et du format interne de chaque composante à mettre à jour (se trouvant dans L)*
- 5) *reconnaissance du type d'opération : modification, ajoute ou suppression*
 entrez la valeur de la clé, et du reste du tuple
 accès aux données réelle grâce à l'adresse physique (ADRPHY)
 si la valeur de la clé existe dans le fichier contenant la relation
 aller en 6 (il s'agit d'une modification ou d'une ajoute)
 sinon aller en 7 (il s'agit d'une suppression)
- 6) *modification des données d'un tuple de la relation*
 entrez l'indicateur de suppression
 si INDIC = 0 alors modification des valeurs réelles avec contrôle du
 type et du format de chaque composante à modifier
 aller en 10
 si INDIC = 1 alors archivage du tuple
 suppression du tuple
 modification dans VERSIONS de l'état (éventuellement)
 et du cardinal (CARDIN) correspondant à la relation

- 7) *ajoute d' un tuple à la relation après le TEST DE LA PRESENCE OBLIGATOIRE* (cfr. 5.5.2) qui consiste en :
 grâce à keyid trouvé par (relogid,keyid) \in (RELOGID,KEYID) RELLOG
 on accède à la relation KEYUNIT
 si (keyid,composante non temporelle 1,composante non temporelle 2 de la clé) \in (KEYID,UNITID 1,UNITID 2) KEYUNIT aller en 8
 sinon aller en 9
- 8) *les valeurs des unitid sélectionnés en 7 doivent appartenir au noyau*
 si les valeurs des unitid sélectionnés en 7 appartiennent au noyau
 alors aller en 9
 sinon "création de tuple interdite car absence de données signalétiques"
 aller en 17
- 9) *ajoute effective du tuple à la relation physique*
 ajoute du tuple à la relation avec contrôle du type et du format interne de chaque composante ajoutée
 NBOCC = 1 (à chaque ajoute de tuple, il est initialisé à 0)
- 10) *y a-t-il des données à calculer ?*
 soit $c \in L$ (liste des composantes à mettre à jour)
 ✓ c , si $(c,'p') \in$ (DATAID,PRIMAIRE) DATATYPE aller en 11
 sinon aller en 15
- 11) *y aura-t-il des calculs à faire ?*
 grâce aux dataid sélectionnés en 10 accès à COMDATA
 nous possédons les numdata des dataid grâce à leur sélection dans COMPOSANT de la manière suivante :
 $(relphid,dataid,numdata) \in$ (RELPHID,DATAID,NUMDATA) COMPOSANT
 si (numdata) \in (FORMULE) COMDATA aller en 12
 sinon aller en 15 (il ne s' agit pas d' une donnée primaire)
- 12) *la formule comportant un ensemble de numdata il s' agit de retrouver leur dataid*
 ✓ numdata de la formule de calcul sélectionnés en 11, rechercher leur dataid : $(relphid,dataid,numdata) \in$ (RELPHID,DATAID,NUMDATA) COMPOSANT
- 13) *dispose-t-on de toutes les données pour effectuer le calcul ?*
 en d' autres termes : la période de référence des données intervenant dans le calcul doit être la même.
 soit P l' ensemble des périodes de références des dataid sélectionnés
 ✓ $p \in P$, si $(relphid,période \text{ de ref}) \in$ (RELPHID,PREF) VERSIONS
 alors aller en 14
 sinon aller en 15

- 14) *puisque' on dispose de la même version de données pour toutes les composantes intervenant dans le calcul, on peut exécuter celui-ci*
mise à jour de la donnée calculée grâce au programme de calcul
- 15) *mise à jour éventuelle des informations concernant la version de la relation mise à jour*
s' il n' y a pas eu création d' un tuple aller en 17
si (relphid,période de référence) \in (RELPHID,PREF) VERSIONS
alors $NBOCC = NBOCC + \text{cardin}$ (cardin appartient à VERSIONS)
et aller en 16
sinon nous savons que $NBOCC = 1$
- 16) *mise à jour dans VERSIONS du tuple correspondant à la relation*
on recherche le cardinal de keyunit correspondant à la relation
si $NBOCC = \text{cardinal}$ (composante de KEYUNIT)
la version est complète
le tuple (relphid,pref,'c',NBOCC) caractérisera la relation dans
VERSIONS pour cette période de référence
si $NBOCC < \text{cardinal}$
la version est incomplète
le tuple (relphid,pref,'p',NBOCC) caractérisera la relation dans
VERSIONS pour cette période de référence
si $NBOCC > \text{cardinal}$
impossible si le test de présence obligatoire a été effectué
lors du comptage de NBOCC
- 17) *y a-t-il encore des tuples à mettre à jour' ?*
si la mise à jour est terminée stop
sinon aller en 5

5.5.1.2 Archivage des données de l' entrepôt

L' entrepôt des valeurs ne peut conserver indéfiniment les données accumulées au fil du temps. Periodiquement, il s' agit d' archiver les données afin de conserver les valeurs les plus anciennes. La composante NBMXVR de la relation RELLOG nous fournira le délai de détention maximum pour les données de type 'événement' et le nombre maximum de versions à conserver par le système pour les données de type 'situation'.

5.5.1.2.1 Archivage des données de caractéristique de mouvement 'événement'

Les différentes étapes de l' archivage des données de type 'situation' sont les suivantes :

- rechercher les relations logiques ayant une caractéristique de mouvement 'événement'

- pour chaque relation logique, rechercher ses relations physiques
- vérifier s' il y a lieu d' archiver grâce à la date de détention maximum
- archiver les tuples les plus anciens dans un fichier d' archivage

1) *recherche des relations logiques ayant comme caractéristique de mouvement 'événement'*

balayer la relation RELLOG de la base de données descriptive et sélectionner les relations telles que :

$(relogid, 'ev') \in (RELOGID, MVT) \text{ RELLOG}$

si aucune relation n' est sélectionnée stop

2) *recherche des relations physiques des relations logiques sélectionnées en 1*
prendre les relations physiques et leur adresse physique

$(relphid, adrphy, relogid) \in (RELPHID, ADRPHY, RELOGID) \text{ RELPHY}$

3) *le délai de détention maximum est-t-il dépassé ?*

notons que la date d' observation peut faire partie de la clé pour les données de type 'événement' (cfr. 5.1.3.1)

on accède aux données réelles grâce à l' adresse physique (ADRPHY)

si la date d' observation < date du jour - délai de détention maximum

alors aller en 4

sinon aller en 5

4) archivage des tuples les plus anciens et suppression de ceux-ci dans le fichier des données réelles

5) *y a-t-il encore des relations à archiver ?*

si oui aller en 3

sinon stop

5.5.1.2.2 Archivage des données de caractéristique de mouvement 'situation'

Rappelons que le nombre maximum de versions conservées dans l'entrepôt est connu grâce à la composante NBMXVR de la relation VERSIONS.

Le nombre de versions d'une relation physique est modifié lors de l'ajoute d'une nouvelle version, il y a normalement lieu de vérifier si un archivage ne doit pas être effectué en contrôlant si le nombre maximum de versions n'est pas atteint pour une relation physique.

Nous proposons, plutôt que d'effectuer un éventuel archivage à chaque ajout d'une version, d'effectuer un archivage périodique tout comme pour les données caractérisées 'événement'.

La période séparant deux archivages sera calculée en fonction de la dimension de référence des relations logiques c'est à dire, on archivera plus souvent les relations ayant une période de référence courte (mois) que les relations ayant une période de référence plus longue (année par exemple).

Les différentes étapes de l'archivage de données 'situation' sont les suivantes : - rechercher les relations logiques ayant comme caractéristique de mouvement 'situation'

- pour chacune de ces relations logiques, prendre ses relations physiques
- vérifier s'il y a lieu d'archiver
- archiver les tuples les plus anciens dans un fichier d'archivage

1) *recherche des relations logiques ayant comme caractéristique de mouvement 'situation'*

balayer la relation RELLOG de la B.D.D. et sélectionner les relations telles que : $(relogid, 'sit') \in (RELOGID, MVT) RELLOG$
si aucune relation n'est sélectionnée stop

2) *recherche des relations physiques des relations logiques sélectionnées en 1*
prendre les relations physiques et leur adresse physique

$(relphid, adrphy, relogid) \in (RELPHID, ADRPHY, RELOGID) RELPHY$

3) *le nombre maximum de version est-il atteint ?*

on accède à la relation VERSIONS et on compte le nombre de versions différentes trouvées pour chaque relation physique sélectionnée en 2

soit x le nombre trouvé,

si $x \leq NBMXVR$ aller en 4 (il ne faut pas encore archiver)

si $x > NBMXVR$ alors : - supprimer les tuples correspondant aux versions à archiver, dans VERSIONS
- archiver les tuples les plus anciens
- supprimer ces tuples du fichier réel

- 4) y a-t-il encore des relations à archiver ?
 si oui aller en 3
 sinon stop

5.5.2 Contrainte de la présence obligatoire

Afin d'assurer un certain contrôle sur les données contenues dans l'entrepôt, une contrainte de présence obligatoire peut être imposée avant de créer des données, en effet, pour créer une unité on doit avoir enregistré l'identification de celle-ci dans le noyau.

Lors de la création de tuples pour une relation, il faut, pour chaque unité non temporelle de la clé primaire, vérifier si elle appartient au noyau. Si c'est le cas, les valeurs des unités à créer devront être contenues dans le noyau. Nous reviendrons plus en détail sur le contenu du noyau (cfr. 6.2.2) et nous verrons que nous avons fait certains choix.

5.5.3 Versions partielles et complètes

Lors de la création de données de caractéristique de mouvement 'situation' il y a lieu de vérifier la dimension de référence de la relation concernée. Rappelons que cette dimension de référence est trouvée automatiquement car elle fait partie de la clé primaire; elle correspond aux composantes temporelles de cette clé.

Deux cas peuvent alors se présenter :

- a) on ajoute des données à une version incomplète qui peut ainsi devenir complète c'est à dire : il faut que le nombre d'occurrences comptées pour la version considérée soit égal au cardinal correspondant à l'identificateur du type de clé de cette relation. Le tuple correspondant à la relation mise à jour, est modifié : l'état de 'p' passe à 'c' et le cardinal de cette version est égal au cardinal de l'identificateur du type de clé de cette relation.
- b) on ajoute des données pour une version qui n'existait pas encore et qui devient soit incomplète soit complète (pour cette dernière possibilité cfr. ci-dessus).
 Si la version est incomplète c'est que le nombre d'occurrences comptées pour la version considérée est plus petit que le cardinal correspondant à l'identificateur du type de clé de cette relation.

Le tuple (relation phys., période de référence, 'c', nombre d' occ. comptées)
est ajouté à la relation VERSIONS si la version est complète.

Le tuple (relation phys., période de référence, 'p', nombre d' occ. comptées)
est ajouté à la relation VERSIONS si la version est incomplète.

5.5.4 Données calculées

Explicitons la règle choisie par (BDER) pour déterminer à quel moment effectuer un calcul : pour une donnée calculée, le calcul sera effectué après l'opération de mise à jour (et par conséquent le résultat sera stocké dans une des relations de l'entrepôt dont le critère de calcul est égal à 1) si les éléments du calcul figurent dans des relations logiques différentes. Si les éléments se trouvent dans la même relation logique, le calcul se fera au moment de la consultation et sera donc effectué à chaque consultation. En effet, au moment de la consultation, l'utilisateur accède à l'entité de base en consultation, la relation logique. Pour le système, une relation logique est un ensemble de relations physiques qu'il doit accéder en même temps. Dès lors, si tous les éléments du calcul sont des composantes de la relation logique, aucun accès supplémentaire n'est nécessaire et il n'y a pas de longue attente pour le montage de volumes, etc...

5.5.5 Mise à jour de la base de données descriptive ou extensibilité du modèle

Le gestionnaire de la base de données a la possibilité d'effectuer des opérations de mise à jour de la base de données descriptive (B.D.D.): ajoute, modification ou suppression. Cette éventualité garantit l'extensibilité de la base de données. La base de données économique régionale, composée d'un ensemble de relations, est un ensemble mouvant. On conçoit, en effet, de devoir ajouter ou supprimer une relation logique ou physique, de même, ajouter ou supprimer une composante d'une relation physique si les besoins s'en faisaient sentir. Ainsi, à l'intérieur d'une relation logique, le nombre des composantes des relations physiques la formant, est susceptible de subir des modifications au cours de la vie de la base de données et de provoquer une restructuration du fichier de chaque relation physique modifiée. Il est donc souhaitable de posséder un modèle ayant une extensibilité maximale. Nous devons posséder un interface général (peu importe son langage d'implémentation) capable de modifier automatiquement les relations de la base de données descriptive concernées.

Nous envisageons six types d' extension :

- l' ajoute d' une relation logique
- la suppression d' une relation logique
- l' ajoute d' une relation physique
- la suppression d' une relation physique
- l' ajoute d' une composante à une relation physique
- la suppression d' une composante d' une relation physique

Pour ces types d' extension nous exposerons :

- les informations à fournir par le gestionnaire de la base de données
- les relations manipulées par la base de données descriptive
- l' algorithme à suivre
- les répercussions éventuelles sur les données réelles

5.5.5.1 Ajoute d' une relation logique

Le gestionnaire de la base de données aura les informations suivantes à fournir :

- 5.5.5.1.1
- a) le nom de la relation logique
 - b) le nom des applications qui exploitent les données après consultation de la B.D.
le nom de l' utilisateur de chaque application
la description en clair des applications
 - c) le libellé en clair de la relation logique
 - d) la dimension de référence
 - e) le mot de passe en lecture
 - f) la caractéristique de mouvement
 - g) le nombre maximum de versions
 - h) l' identificateur du type de clé
si l' identificateur du type de clé n' existe pas dans la B.D.D. il devra en plus fournir i) et j)
 - i) les types d' unité faisant partie de la clé
le nombre d' occurrences des relations ayant ce type de clé comme unité-clé
l' identificateur du type de données correspondant
la cardinalité de ces types d' unités dans le système
 - j) les identificateurs de domaines manquants auxquels appartiennent les deux types d' unités
leur type, format interne et externe

Notons que la relation logique ne sera réellement créée qu' après la création de toutes les relations physiques la composant.

5.5.5.1.2 Informations manipulées par la B.D.D.

Notons que les lettres minuscules à droite des tableaux font référence aux informations fournies (cfr. 5.5.5.1.1).

APPLICATION

application	b
relogid	a
userid	b
description	b

(KEYUNIT)

keyid	h
unitid 1	i
unitid 2	i
cardinal	i

RELLOG

relogid	a
titlog	c
dimref	d
mvt	f
nbmxvr	g
keyid	h
readpass	e

(STAND)

domid	j
type	j
formint	j
formext	j

(UNITTYPE)

unitid	i
cardinal	i
dataid	i

5.5.5.1.3 Algorithmme à suivre

Les étapes à suivre lors de l' ajoute d' une relation logique sont les suivantes :

- mise à jour de la relation APPLICATION
- mise à jour de la relation RELLOG
- mise à jour éventuelle de la relation KEYUNIT
- mise à jour éventuelle de la relation UNITTYPE
- mise à jour éventuelle de la relation STAND
- appel de la procédure d' ajoute des relations physiques la composant

Grâce aux informations fournies par le gestionnaire de la base de données :

- 1) ajoute du tuple (application,relogid,userid,description) à la relation APPLICATION pour chaque application définie
- 2) ajoute du tuple (relogid,titlog,dimref,mvt,keyid,readpass) à la relation RELLOG

- 3) si (keyid) \in (KEYID) KEYUNIT aller en 6
sinon ajoute du tuple (keyid,unitid 1,unitid 2,cardinal) à la relation
KEYUNIT
- 4) ajoute pour chaque unitid manquant ajoute du tuple
(unitid,cardinal,dataid) à la relation UNITTYPE
- 5) si le domaine des unitid manquant n' existe pas ajoute, à la relation STAND
pour chaque domaine manquant, du tuple (domid,type,formint,formext)
- 6) pour chaque relation physique contenue dans la relation logique, appel de
la routine d' ajoute d' une relation physique (cfr. 5.5.5.3)
- 7) la relation logique est maintenant réellement créée.

5.5.5.2 Suppression d' une relation logique

5.5.5.2.1 Informations à fournir :

L' unique information à fournir par le gestionnaire de la base de données est le nom de la relation logique . Notons que le gestionnaire supprimant une relation logique doit être conscient de la suppression des relations physiques, des composantes des relations physiques formant cette relation logique.

5.5.5.2.2 Algorithme à suivre

Les étapes à suivre lors de la suppression d' une relation logique sont les suivantes :

- mise à jour de la relation APPLICATION
 - mise à jour de la relation RELLOG
 - mise à jour de la relation KEYUNIT
 - mise à jour éventuelle de la relation UNITTYPE
 - appel de la procédure de suppression pour les relations physiques composant la relation logique
- 1) supprimer de la relation APPLICATION les tuples pour lesquels :
(application,relolid) \in (APPLIC,RELOGID) APPLICATION
 - 2) supprimer de RELLOG après avoir mémorisé keyid le tuple pour lequel :
(relolid) \in (RELOGID) RELLOG
 - 3) *keyid est-il l' identificateur du type de clé à' une autre relation logique ?*
si (keyid) \in (KEYID) RELLOG aller en 5
sinon supprimer de KEYUNIT après avoir mémorisé unitid le tuple
(keyid) \in (KEYID) KEYUNIT

- 4) les types d'unités mémorisés en 3 appartiennent-ils à d'autres types de clé ?
 si (unitid) \in (UNITID 1) KEYUNIT ou (unitid) \in (UNITID 2) KEYUNIT aller en 5
 sinon supprimer dans UNITTYPE les tuples tels que
 (unitid) \in (UNITID) UNITTYPE après avoir mémorisé leur dataid
- 5) recherche des relations physiques de la relation logique à supprimer
 balayer RELPHY et mettre dans la Table des relations physiques à supprimer
 tous les relphid pour lesquels
 (relphid, relogid) \in (RELPHID, RELOGID) RELPHY
- 6) mettre les dataid mémorisés en 4 dans la Table des composantes à supprimer
 pour chacun d'eux appel de la procédure de suppression d'une composante
- 7) pour chaque élément de la Table des relations physiques à supprimer appel
 de la procédure de suppression d'une relation physique

5.5.5.2.3 Informations manipulées par la B.D.D.

Les informations manipulées par la B.D.D. lors de la suppression d'une relation logique sont identiques à celles manipulées lors de l'ajoute d'une relation logique (cfr. 5.5.5.1.2)

5.5.5.3 Ajoute d'une relation physique

5.5.5.3.1 Informations à fournir

Le gestionnaire de la base de données aura à fournir les informations suivantes :

- a) le nom de la relation physique
- b) le nom de l'utilisateur ayant la permission d'accès à cette relation physique
- c) le niveau d'agrégation autorisé
- d) le libellé en clair de la relation physique
- e) l'adresse physique de la relation
- f) la fréquence de collecte
- g) le mot de passe en écriture
- h) le nom de la relation logique à laquelle appartient cette relation physique (ce nom est déjà connu dans le cadre de l'ajoute d'une relation logique)
- i) valeur binaire concernant la "divulgabilité" des données de cette relation
- j) valeur du critère de calcul

5.5.5.3.2 Informations manipulées par la B.D.D.

Notons que les lettres minuscules à droite des tableaux font référence aux informations fournies (cfr. 5.5.5.2.1).

RELPHY

relphid	a
titphy	d
adrphy	e
userid	b
fcoll	f
wypass	g
relogid	h
divulg	i
calculée	j

PERMISSION

userid	b
relphid	a
agreg	c

5.5.5.3.3 Algorithme à suivre

Les étapes à suivre lors de l'ajoute d'une relation physique sont les suivantes :

- mise à jour de la relation PERMISSION
- mise à jour de la relation RELPHY
- appel de la procédure d'ajoute des composantes pour toutes les composantes qu'elle contiendra

Grâce aux informations fournies par le gestionnaire de la base de données :

- 1) ajoute du tuple (userid,relphid,agreg) à la relation PERMISSION
- 2) ajoute du tuple (relphid,titphy,adrphy,userid,fcoll,wypass,relolid,divulg,calculée) à la relation RELPHY
- 3) pour chaque composante qui appartiendra à la relation physique appel de la routine d'ajoute d'une composante à une relation physique

Notons que pour créer une relation physique, le gestionnaire doit d'abord créer la relation logique à laquelle celle-ci appartiendra.

5.5.5.4 Suppression d'une relation physique

5.5.5.4.1 Information à fournir

La seule information à fournir par le gestionnaire est le nom de la relation physique.

Contrainte : Le gestionnaire supprimant une relation physique ne peut le faire que si le nombre de relations physiques appartenant à une relation logique est plus grand que 1, sinon cela reviendrait à supprimer une relation logique.

5.5.5.4.2 Informations manipulées par la B.D.D.

Les informations manipulées par la B.D.D. lors de la suppression d' une relation physique sont, outre les informations manipulées lors de l' ajoute d' une relation physique (cfr. 5.5.5.3.2), les informations des relations VERSIONS, APPLIC et COMPOSANT.

5.5.5.4.3 Algorithme à suivre

Les étapes à suivre lors de la suppression d' une relation physique sont les suivantes :

- mise à jour de la relation VERSIONS
- mise à jour de la relation PERMISSION
- mise à jour de la relation RELPHY
- mise à jour éventuelle de la relation APPLICATION
- mise à jour de la relation COMPOSANT
- appel de la routine de suppression d' une composante

- 1) supprimer de la relation VERSIONS les tuples pour lesquels :
 $(relphid) \in (RELPHID) \text{ VERSIONS}$
- 2) Si cela n' a pas été fait dans le cadre de la suppression d' une relation logique, supprimer de RELPHY le tuple tel que $(relphy) \in (RELPHID) \text{ RELPHY}$
- 3) balayer la relation PERMISSION et supprimer le tuple pour lequel
 $(userid, relphid) \in (USERID, RELPHID) \text{ PERMISSION}$
 si $(userid, relphid) \notin (USERID, RELPHID) \text{ PERMISSION}$ et
 $(userid, relphid) \notin (USERID, RELPHID) \text{ RELPHY}$ l' utilisateur n' a plus de raison d' être connu du système il peut être définitivement rayé en supprimant de APPLICATION les tuples tels que $(userid) \in (USERID) \text{ APPLICATION}$
- 4) balayer COMPOSANT et supprimer (après avoir mémorisé les dataid et numdata dans la table des composantes à supprimer) les tuples tels que
 $(relphid) \in (RELPHID) \text{ COMPOSANT}$
- 5) pour chaque élément de la table des composantes à supprimer appeler la routine de suppression d' une composante d' une relation physique.

5.5.5.4.4 Répercussions sur les données réelles

Avant de supprimer le fichier des données réelles on archive celui-ci. Il en va de même pour les informations de gestion contenues dans la base de données descriptive (cfr. 5.5.5.5.4 et 5.5.5.6.4)

5.5.5.5 Ajoute d' une composante à une relation physique

5.5.5.5.1 Informations à fournir

Le gestionnaire de la base de données aura à fournir les informations suivantes :

- a) le nom de la relation physique à laquelle la composante va appartenir (sauf dans le cadre de la création d' une relation physique)
- b) le nom de la composante
- c) le numéro de la composante
- d) le libellé en clair de la composante
- e) l' unité de mesure de la composante
- f) l' indicateur de sa possibilité de saisie
- g) déplacement de la composante par rapport au début du fichier
- h) l' identificateur du type de domaine auquel appartient la composante
si cet identificateur du type de domaine n' existe pas encore dans la B.D.D., il devra en plus fournir i)
- i) le type de cet identificateur, son format interne et externe
- j) la caractéristique primaire de la composante
si elle est 'c' ou 'd' il fournira en plus k)
- k) la formule de calcul servant à calculer cette composante
le nom du programme de calcul calculant celle-ci

5.5.5.5.2 Informations manipulées par la B.D.D.

Notons que les lettres minuscules à droite des tableaux font référence aux informations fournies (cfr ci-dessus)

DATATYPE

dataid	b
domid	h
primaire	j

(COMDATA)

dataid	b
formule	k
progr	k

COMPOSANT

relphid	a
dataid	b
numdata	c
titdata	d
mesure	e
possai	f
déplacement	g

(STAND)

domid	h
type	i
formint	i
formext	i

5.5.5.5.3 Algorithme à suivre

Les étapes à suivre lors de l'ajoute d'une composante à une relation physique sont les suivantes :

- mise à jour de la relation COMPOSANT
- mise à jour de la relation DATATYPE
- mise à jour éventuelle de la relation STAND
- mise à jour éventuelle de la relation COMDATA

Contrainte d'intégrité : le nom de la composante à ajouter ne peut pas encore appartenir à la relation logique

Grâce aux informations fournies par le gestionnaire de la base de données :

- 1) ajoute du tuple (relphid,dataid,numdata;titdata,mesure,possai,déplacement) à la relation COMPOSANT
- 2) ajoute du tuple (dataid,domid,primaire) à la relation DATATYPE
 si (domid) \in (DOMID) STAND aller en 3
 sinon ajoute du tuple (domid,type,formint,formext) à la relation STAND
- 3) si primaire = 'r' (la composante n'intervient pas dans un calcul) aller en 4
 sinon ajoute du tuple (dataid,formule,progr) à la relation COMDATA
lors de cette ajoute le gestionnaire veillera à la cohérence de la base de données (le programme de calcul doit exister,...)

5.5.5.5.4 Répercussion sur les données réelles

Ces répercussions ne sont valables que dans le cadre de l'ajoute d'une composante à une relation déjà existante.

Nous allons examiner les cas suivants : soit le fichier d'archivage des données réelles n'est pas vide il faut alors

- reformater le fichier des données réelles
- tenir compte des répercussions sur la base de données descript.
- créer un nouveau fichier d'archivage,

soit le fichier d' archivage des données réelles est vide il suffit alors

- de reformater le fichier des données réelles
- de tenir compte des répercussions sur la base de données descriptive.

Notons que l' ajout d' une composante se fait toujours à la droite des autres composantes de la relation physique existante afin d' opérer un minimum de modification des données de gestion (déplacement des composantes par exemple).

Nous proposons d' ajouter 2 composantes à la relation RELPHY :

- LONGUEUR : longueur du fichier contenant la relation physique
- DATDEBVAL : date de début de validité : date à partir de laquelle la nouvelle description est valable

Algorithme :

- 1) si le fichier d' archivage correspondant à la relation physique n' est pas vide il faut :
 - reformater le fichier des données réelles (c' est à dire agrandir sa longueur)
 - archiver le tuple de RELPHY correspondant à relphid avec l' ancienne longueur et la date de fin de validité de la description afin de pouvoir éventuellement retrouver l' ancienne description.
 - créer un nouveau fichier d' archivage (car il ne servirait pas à grand chose de reformater un fichier qui n' est pas destiné à être consulté par les utilisateurs, de plus, le coût en serait assez élevé étant donné le nombre de données)
- 2) mettre à jour le tuple correspondant à relphid dans RELPHY (nouvelle longueur et nouveau déplacement)
- 3) si le fichier d' archivage est vide il suffit de :
 - reformater le fichier contenant la relation physique

Notons que la place perdue , pour les anciennes versions, occasionnée par l' ajout de la nouvelle composante sera remplacée par une valeur conventionnellement nulle. Si le tuple de RELPHY archivé contenait en plus de la date de début de validité (correspondant à la date d' archivage) une date de fin de validité, on pourrait retrouver plus facilement la période de validité de cette description de la relation physique.

5.5.5.6 Suppression d' une composante d' une relation physique

5.5.5.6.1 Informations à fournir

Les informations à fournir par le gestionnaire de la base de données sont les suivantes :

- le nom de la composante
- le nom de la relation physique à laquelle cette composante appartient (sauf dans le cadre de la suppression d' une relation physique)

Contraintes :

- le gestionnaire supprimant une composante d' une relation ne pourra le faire que si le nombre de composantes non primaires est plus grand que 1. Dans le cas contraire il s' agirait de la suppression de la relation physique.
- si la composante supprimée a comme critère de calcul 'p' (primaire) ou 'd' (calculée et primaire) il y a lieu de vérifier si cette suppression n' entraîne pas l' altération d' une formule de calcul.
- si composante supprimée a comme critère de calcul 'c' (calculée) ou 'd' (calculée et primaire) il y a lieu de vérifier si cette suppression n' entraîne pas la modification du critère de calcul de composantes 'primaires' au cas où elles ne seraient utilisées que dans le calcul de cette composante.

5.5.5.6.2 Informations manipulées par la B.D.D.

Les informations manipulées par la B.D.D. lors de la suppression d' une composante d' une relation physique sont identiques à celles manipulées lors de l' ajout d' une composante à une relation physique. (cfr. 5.5.5.5.2)

5.5.5.6.3 Algorithme à suivre

Les étapes à suivre lors de la suppression d' une composante d' une relation physique sont les suivantes :

- mise à jour éventuelle de la relation COMPOSANT
- mise à jour de la relation DATATYPE
- mise à jour éventuelle de la relation COMDATA
- mise à jour éventuelle de la relation STAND

1) si cela n' a pas été fait dans le cadre de la suppression de la relation physique : supprimer de la relation COMPOSANT le tuple tel que :

(relphid,dataid) \in (RELPHID,DATAID) COMPOSANT

- 2) si (dataid) \notin (DATAID) COMDATA aller en 3
 sinon-mémoriser tous les numdata appartenant à une formule ainsi que leur dataid correspondant
 -supprimer de COMDATA le tuple tel que (dataid) \in (DATAID) COMDATA
la suppression d' une donnée calculée entraîne celle du programme de calcul correspondant
- 3) supprimer de DATATYPE le tuple tel que : (dataid) \in (DATAID) DATATYPE
 si le domid de dataid supprimé en 3 n' appartient plus à DATATYPE il faut supprimer dans STAND le tuple tel que (domid) \in (DOMID) STAND
 sinon rien
- 4) pour les numdata mémorisés en 2
 si (numdata) \in (FORMULE) COMDATA aller en 6
 sinon *la donnée n' intervient plus dans aucun calcul*
- 5) pour les numdata et dataid mémorisés en 2
 si (dataid,'p') \in (DATAID,PRIMAIRE) DATATYPE remplacer la valeur de la composante PRIMAIRE par 'r'
 si (dataid,'d') \in (DATAID,PRIMAIRE) DATATYPE remplacer la valeur de la composante PRIMAIRE par 'c'
- 6) fin de l' ajoute d' une composante à une relation physique

5.5.5.6.4 Répercussions sur les données réelles

Examinons les deux cas possibles de répercussions :

a) dans le cadre de la suppression d' une relation physique

Après avoir supprimé toutes les composantes de cette relation,

- on archive le fichier contenant la relation physique
- on détruit le fichier contenant la relation physique
- on archive les informations de gestion supprimées dans la base de données descriptive (afin de pouvoir éventuellement revenir en arrière.

b) dans le cadre d' une suppression d' une composante d' une relation physique qui continuera d' exister

- la longueur du fichier des données réelles reste provisoirement intacte
- lors de l' ajoute de nouvelles versions, à la place de la composante supprimée on indique une valeur conventionnellement neutre
- on archives les informations de gestion supprimées dans la base de données descriptive.

Périodiquement, pour les fichiers contenant des relations ayant eu des composantes supprimées, il faut vérifier s' il existe encore des données concernant les composantes supprimées :

s' il n' en existe plus on exécute les opérations suivantes :

- reformater le fichier des données réelles
- archiver le tuple de RELPHY avec l' ancienne longueur et la date de fin de validité
- on crée un nouveau fichier d' archivage (selon le nouveau format)
- on modifie le tuple correspondant à la relation dans RELPHY : on indique la nouvelle longueur et la date de début de validité.

sinon il n' y a rien à faire car il existe encore des données de l' ancienne description.

5.5.6 Mise à jour massive de l' entrepôt des valeurs

La mise à jour massive consiste à apporter une nouvelle version à une relation physique de l' entrepôt. L' ajoute de nouvelles versions suit le même principe que celui de la création de la base de données (cfr. 5.2) c' est à dire la copie d' un fichier contenant la version d' une relation dont la caractéristique de mouvement est 'événement' ou 'situation'.

Nous parlons d' ajoute de nouvelles versions car il semblerait utile de conseiller à l' utilisateur, vu les coûts élevés de manipulation, d' ajouter des versions complètes. Néanmoins, s' il le désire, il pourrait ajouter des versions incomplètes.

5.5.6.1 Format général de la commande

MAJM relation,writepass,période de référence

où MAJM : mot clé correspondant à l' opération de mise à jour massive

relation : relation physique à mettre à jour

writepass : mot de passe en écriture de la relation

période de référence : localisation temporelle des données de cette relation

5.5.6.2 Algorithme de la mise à jour massive

Les contrôles suivants sont effectués lors de la mise à jour massive des relations de l'entrepôt :

- vérification syntaxique des paramètres de la commande
- contrôle de l'existence de la relation à mettre à jour
- vérification du critère de calcul de la relation
- vérification du mot de passe de l'utilisateur
- de plus, il y a appel de la routine de mise à jour massive de données caractérisées 'événement' ou 'situation' où d'autres contrôles sont effectués.

Pour les points a), b), c), d) et e) nous renvoyons à la mise à jour ponctuelle en 5.5.1.1.2 points a), b), c), d) et e).

- a) *contrôle syntaxique des paramètres de la commande*
- b) *vérification de l'existence de la relation*
- c) *vérification du critère de calcul*
- d) *l'utilisateur a-t-il le droit de mettre à jour cette relation ?*
- e) *l'utilisateur a-t-il fourni le bon mot de passe correspondant à cette relation ?*
- f) *détermination de la caractéristique de mouvement de la relation*

grâce à relation rechercher dans RELLOG la relation logique (relogid)
à laquelle appartient la relation physique
si (relogid,'ev') \in (RELOGID,MVT) RELLOG aller en A)
si (relogid,'sit') \in (RELOGID,MVT) RELLOG aller en B)
sinon " mise à jour d'une relation n'appartenant pas à l'entrepôt" stop

- A) Algorithme de mise à jour massive de données de type 'événement'
-

En plus des contrôles effectués en 5.5.6.2 d'autres vérifications sont effectuées :

- contrôle du type et du format de chaque composante
- vérification de la contrainte de présence obligatoire

- 1) sur base de relphid on accède à COMPOSANT et on sélectionne tous les dataid tels que : (relphid,dataid) \in (RELPHID,DATAID) COMPOSANT

- 2) *recherche du type et du format interne de chaque composante*
pour tous les dataid sélectionnés en 1 rechercher leur domaine
 $(dataid, domid) \in (DATAID, DOMID) DATATYPE$
grâce à ces domid on trouve le type et le format des composantes :
 $(domid, type, formint) \in (DOMID, TYPE, FORMINT) STAND$
- 3) *contrainte de la présence obligatoire*
grâce à la keyid de la relation accès à la relation KEYUNIT
si $(keyid, unitid\ 1, unitid\ 2) \in (KEYID, UNITID\ 1, UNITID\ 2)$ aller en 4
sinon lire un tuple de la version à ajouter
aller en 5
- 4) *les valeurs des unitid sélectionnés en 3 doivent appartenir au noyau*
lecture d' un tuple de la version à ajouter
si la valeur des unitid sélectionnés en 3 appartient au noyau aller en 5
sinon "données signalétiques absentes pour les unitid" aller en 6
- 5) *ajoute d' un tuple*
ajoute, en fin de fichier contenant la relation physique, d' un tuple
- 6) *est-ce la fin de la mise à jour massive ?*
y a-t-il encore des tuples à lire ?
si oui aller en 4
sinon stop

B) Algorithme de mise à jour massive de données de type 'situation'

En plus des contrôles effectués en 5.5.6.2 d' autres vérifications sont effectuées :
- contrôle du type et du format des composantes
- contrôle de l' existence de la version
- vérification de la contrainte de présence obligatoire

- 1) sur base de relphid on accède à COMPOSANT et on sélectionne tous les dataid tels que : $(relphid, dataid) \in (RELPHID, DATAID) COMPOSANT$
- 2) *recherche du type et du format interne de chaque composante*
pour tous les dataid sélectionnés en 1 rechercher leur domaine
 $(dataid, domid) \in (DATAID, DOMID) DATATYPE$
grâce à ces domid on trouve le type et le format des composantes :
 $(domid, type, formint) \in (DOMID, TYPE, FORMINT) STAND$

- ```

3) la version est-elle connue du système ?
 si (relation,période de référence) ∈ (RELPHID,PREF) VERSIONS aller en 7
 sinon aller en 4

4) la version est inconnue du système
 lecture d' un tuple
 test de la présence obligatoire :
 grâce à keyid trouvé par (relogid,keyid) ∈ (RELOGID,KEYID) RELLOG
 on accède à la relation KEYUNIT
 si (keyid,unitid 1,unitid 2) ∉ (KEYID,UNITID 1,UNITID 2) KEYUNIT aller en 5
 sinon les valeurs des unitid sélectionnés doivent appartenir au noyau
 si elles appartiennent au noyau NBOCC = NBOCC + 1
 aller en 5
 si elles n' appartiennent pas au noyau "creation interdite"
 stop

5) ajoute d' un tuple et vérification d' un éventuel calcul à faire
 ajoute d' un tuple en fin de fichier

5') si il y a des calculs on les effectue (cfr m-a-j ponctuelle)
 sinon aller en 7

6) la version est connue du système
 lecture d' un tuple
 si la valeur de la clé existe sur le fichier des données réelles
 remplacer l' ancien tuple par le nouveau aller en 5')
 sinon effectuer le test de la présence obligatoire
 aller en 5')

7) est-ce la fin de la mise à jour massive ?
 s' ils n' y a plus de tuple à lire aller en 8
 si il en a encore aller en 4 ou en 6 selon que la version est connue du
 système ou non

8) mise à jour des informations concernant la version de la relation
 si la version est complète elle sera décrite par le tuple
 (relphid,pref,'c',NBOCC) VERSIONS
 si la version est incomplète elle sera décrite par le tuple
 (relphid,pref,'p',NBOCC) VERSIONS
 (cfr. mise à jour ponctuelle de données 'situation')

```

(cfr. mise à jour ponctuelle de données 'situation')



## 5.6 Problèmes particuliers de la gestion automatique de la base de données

Nous proposons une ébauche de solution de certains problèmes posés par la gestion automatique de la base de données.

### 5.6.1 Les séries chronologiques

Il est possible d'obtenir des séries chronologiques à partir d'une vue diachronique des données en triant les relations sur établissement ou entreprise et le numéro de version. L'utilisateur pourra à sa guise établir des séries chronologiques en triant les données avec comme argument primaire les composantes non temporelles de la clé et comme argument secondaire les composantes temporelles de la clé.

### 5.6.2 Les versions partielles

Nous avons la possibilité de faire des rappels aux organismes-sources pour la collecte des données. Un délai maximum à ne pas dépasser doit être accordé entre la date correspondant à la fréquence de collecte et la date de réception des données.

Périodiquement, on choisit les relations physiques de caractéristique de mouvement 'situation'. Pour toutes ces relations physiques :

si  $(\text{date du jour} - \text{période de référence de la dernière version complète}) >$

$\text{délai} + \text{fréquence de collecte}$  il faudra alors envoyer un rappel aux organismes-sources mis en cause afin de récolter au plus vite les données en retard.

### 5.6.3 Données calculées

Nous proposons de calculer les valeurs des données calculées lors de la mise à jour, ceci afin d'uniformiser les traitements et de gagner du temps lors de la consultation de la base de données. Le nombre de données calculées, à partir d'agrégats ou non, lors de la consultation étant trop faible (voir même exceptionnel) que pour réaliser des traitements spéciaux lors de la consultation. La perte de place causée par la maintenance de ces données étant infime par rapport au reste de la base de données

#### 5.6.4 Le législateur et la base de données

Grâce à l'opération de consultation de la base de données descriptive le législateur pourra obtenir divers renseignements :

- la liste des utilisateurs ayant accès au système
- les applications que les utilisateurs effectuent grâce aux données qu'ils tirent de l'entrepôt
- etc ...

Pour plus de détail nous renvoyons le lecteur à l'approche juridique et socio-politique définie dans (BDER pp. 265-382).

#### 5.6.5 Autres problèmes

Il existe d'autres problèmes tels que, par exemple, celui de la concurrence entre les utilisateurs désirant accéder aux données ou celui des back-up pouvant être précieux lors de la perte accidentelle d'informations, qui ne sont pas pris en considération.



## CHAPITRE 6 IMPLEMENTATION REALISEE LORS DU STAGE

Ce chapitre s'emploie à décrire la réalisation partielle de l'implémentation de la base de données proposée par (BDER). Alors que le chapitre précédent s'attachait à décrire des algorithmes généraux pour la réalisation des applications de mise à jour et de consultation indépendamment de l'outil utilisé, nous décrivons ici celui mis à notre disposition, un descriptif existant que nous avons repris pour des raisons de rapidité. Insistons sur le fait qu'il contient une partie des données de la base de données descriptive mais que son utilisation n'est pas indispensable pour implémenter la base de données si on emploie tous les types de données de la B.D.D..

L'application de mise à jour utilise partiellement la base de données descriptive proposée par (BDER) ainsi que le descriptif existant tandis que l'application de consultation n'emploie pas la B.D.D. mais bien un langage d'interrogation prototype mis à notre disposition.

### 6.1 Description de l'outil utilisé

Pour réaliser l'implémentation physique de la base de données nous disposons de :

- une machine virtuelle
- un système d'exploitation
- un langage de programmation
- un langage d'interrogation de bases de données

#### 6.1.1 Une machine virtuelle

Dans un système à machines virtuelles, chaque utilisateur dispose virtuellement d'un ordinateur. Ce système est destiné à effectuer un partage de temps en simulant pour chaque utilisateur un ordinateur en 'activant' périodiquement cet 'ordinateur'.

#### 6.1.2 Un système d'exploitation

La machine virtuelle que nous utilisons travaillait grâce à un système d'exploitation à partage de ressources : C.M.S. (Central Monitoring System). Nous ajouterons simplement que le C.M.S. est particulièrement adapté au mode conversationnel.

### 6.1.3 Un langage de programmation

Nous avons utilisé A.P.L. comme langage de programmation des différents programmes destinés à créer et mettre à jour la base de données. L'interpréteur A.P.L. permet de faire en interactif une mise à jour ponctuelle de données et nous a permis d'effectuer rapidement, en conversationnel, des tests de validité d'algorithmes. Ceci reste valable même si l'utilisateur n'est pas informaticien averti.

### 6.1.4 Un langage d'interrogation de bases de données

Le langage d'interrogation de bases de données utilisé par l'application de consultation est 'LAPIN' (LAngage Pour Interrogation). Ce langage est un prototype et est décrit dans (LAPIN). L'emploi de celui-ci est détaillé en 6.5.1.

## 6.2 Choix préliminaires

Avant d'aborder l'implémentation de la base de données précisons d'abord certains choix effectués :

- le subset de données
- le noyau
- la dimension de référence

### 6.2.1 Subset de données

L'implémentation réalisée ne s'applique pas à l'entièreté du dictionnaire des données présenté dans (BDER pp. 70-117) mais à un sous-ensemble de celui-ci seulement, notre but étant essentiellement de tester une méthode. La généralité de la méthode est cependant telle qu'il n'y aurait aucune difficulté à l'appliquer à l'ensemble du dictionnaire de données. La liste des données retenues est la suivante : (cfr. BDER p. 177)

- 110001 à 110009 : identifiant de l'entreprise, de l'établissement
- 12000N à 12010N : secteur d'activité de l'entreprise, de l'établ.
- 210001 à 215006 : composantes de la valeur ajoutée
- 221001 à 221109 : structure du prix de revient
- 222001 à 220003 : produits fabriqués
- 240001 à 240008 : pollution
- 311001 à 312113 : emploi
- 313001 à 313010 : productivité
- 315001 à 315107 : formation permanente



321001 à 321011 : stock de capital  
 322001 à 323005 : technologie de production  
 324001 à 324013 : investissement  
 414001 à 414030 : aide à l'investissement  
 421001 à 425016 : données financières  
 440001 à 440007 : difficulté de gestion

Les numéros ci-dessus sont ceux correspondants aux intitulés du dictionnaire d'informations présenté par (BDER PP. 70-117) et dont le lecteur trouvera la description plus détaillée à l'annexe n° 1.

### 6.2.2 Le noyau

Nous avons mis dans le noyau des données signalétiques concernant les établissements, les entreprises et les travailleurs c'est à dire trois relations logiques et sept relations physiques. Leur description se fait comme suit :

Relation IDET1 :  
*id. établissement*

|          |                                                |
|----------|------------------------------------------------|
| NOM      | 110001 Nom de ET                               |
| ADR      | 110002 Adresse de ET                           |
| DATECR   | 110003 Date création ET, date mise en activité |
| EP       | 110004 Identifiant EP dont dépend ET, n° TVA   |
| SECTACTn | 12010N Secteur d'activité N de ET (S'N)        |

Relation IDET2 :  
*id. établissement*

|         |                                 |
|---------|---------------------------------|
| DATEDEX | 210001 Date de début d'exercice |
|---------|---------------------------------|

Relation IDEP :  
*id. entreprise*

|          |                                                   |
|----------|---------------------------------------------------|
| ETHR     | 110005 Identifiant des ET hors région             |
| ADR      | 110006 Adresse EP (siège social)                  |
| NOM      | 110007 Nom ou raison sociale EP                   |
| DATECR   | 110008 Date création EP (publication au Moniteur) |
| FORM     | 110009 Forme juridique EP                         |
| SECTACTn | 12000N Secteur d'activité N de EP (S'N)           |

Relation CARPH5 :  
*res. pot. humain*

|     |                                    |
|-----|------------------------------------|
| NOM | 311101 Nom de l'ET/EP qui l'occupe |
| ADR | 311107 Adresse                     |

Relation CARPH6 :  
*res. pot. humain*

|        |                          |
|--------|--------------------------|
| ID     | 311102 Identité          |
| DNAISS | 311103 Date de naissance |
| SEXE   | 311104 Sexe              |
| NAT    | 311105 Nationalité       |
| ETATC  | 311106 Etat civil        |

Relation CARPH7 :  
*res. pot. humain*

|       |                                             |
|-------|---------------------------------------------|
| TTRAV | 311108 Type de travail                      |
| QDEMP | 301111 Qualification des 5 derniers emplois |

Relation CARPH8 :  
*res. pot. humain*

|       |                                                           |
|-------|-----------------------------------------------------------|
| QSCOL | 311109 Qualification scolaire (Q <sub>1</sub> )           |
| QEMPL | 311110 Qualification de l'emploi occupé (P <sub>1</sub> ) |

Le noyau contient donc des données auxquelles on veut accéder souvent directement car ces informations apportent des renseignements signalétiques. La pratique pourrait nécessiter l'ajoute de relations au noyau lorsqu'elles sont accédées trop fréquemment pour en extraire quelques tuples seulement.

### 6.2.3 La dimension de référence

Notons que la valeur de la dimension de référence des relations logiques est absente de (BDER). Il nous a donc fallu faire un choix arbitraire : nous avons fixé la valeur de la dimension de référence comme suit :

Valeur de la dimension de référence d' une relation logique R =  $\min_{i=1 \text{ à } n} (\text{fréquence de collecte des relations physiques } r_i)$

où  $r_i$  = ième relation physique de R

n = nombre de relations physiques de R

### 6.3 Présentation de notre réalisation

Afin de réaliser l'implémentation de la base de données proposée par (BDER), outre les outils mis à notre disposition (cfr. 6.1) nous avons utilisé la base de données descriptive relationnelle proposée par (BDER) à laquelle nous avons néanmoins apporté quelques modifications : nous disposons d' un descriptif existant contenant déjà des informations de la B.D.D..

Détaillons les optiques suivies pour les deux applications :

#### A) application de consultation

Nous n' avons pas pu, pour l' application consultation, définir notre propre langage d' interrogation de base de données. Pour cette raison, nous employons un prototype (LAPIN) qui se base sur un descriptif existant.

C' est pourquoi ce langage n' utilise pas la base de données descriptive.

#### B) application de mise à jour

L' application mise à jour utilise à la fois la base de données descriptive et les descriptifs des relations existants. Nous avons éliminé de la B.D.D. les informations déjà reprises par le descriptif existant afin de faciliter et d' accélérer l' implémentation de la base de données.

Sur la page suivante nous trouvons un schéma décrivant l' environnement et les outils de gestion des données.



Implémentation physique proposée par (BDER) :

| ENVIRONNEMENT                                                      | GESTION DES DONNEES                                                   | DONNEES                                           |
|--------------------------------------------------------------------|-----------------------------------------------------------------------|---------------------------------------------------|
| <u>outil :</u><br>- ordinateur<br>- op. system<br>- lang. program. | - base de données descriptive<br>relationnelle proposée par<br>(BDER) | <u>base de données :</u><br>- noyau<br>- entrepôt |

Implémentation physique réalisée :

| ENVIRONNEMENT                                                                           | GESTION DES DONNEES                                                                                                                                                                | DONNEES                                           |
|-----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <u>outil :</u><br>- ordinateur<br>- op. system<br>- lang. program.<br>- lang. interrog. | <u>CONSULTATION :</u><br>- descriptif existant<br><br><u>MISE A JOUR :</u><br>- descriptif existant<br>- sous-ensemble de la base de<br>données descriptive<br>proposée par (BDER) | <u>base de données :</u><br>- noyau<br>- entrepôt |

- figure 5 -

Examinons maintenant les descriptifs décrivant les relations ainsi que les informations conservées dans la B.D.D.

### 6.3.1 Le descriptif existant

La description des relations de la base de données est définie grâce à un descriptif general contenant des informations de toutes les relations et un descriptif particulier pour chaque relation dont les images sont les suivantes :

#### 6.3.1.1 Descriptif des relations

Le descriptif général des relations est un tableau (REL) comportant autant de lignes qu'il y a de relations.

Le format général de ce tableau est le suivant :

DESCRIPTIF-GENERAL (RELATION,NBCOMP,ADRPHY,LONGUEUR)

où RELATION : nom de la relation

NBCOMP : nombre de composantes de la relation

ADRPHY : adresse physique où est stockée la relation (nom de fichier)

LONGUEUR : longueur d' un tuple de la relation

Nous avons crée pour chaque relation physique un tuple de ce descriptif.

#### 6.3.1.2 Descriptif des composantes

-----

Il y a autant de descriptifs particuliers que de lignes dans le descriptif général (autant que de relations). Chaque descriptif particulier possède autant de lignes que la relation possède de composantes. Le format général de ces tableaux est le suivant :

DESCRIPTIF-PARTICULIER (COMPOSANTE,RANG,TYPE,FORMEXT,PRECISION,BUFFER,  
DEPLACEMENT,FORMINT,CLE)

où COMPOSANTE : nom d' une composante

RANG : position (rang) de la composante lors de l' édition

TYPE : type de la composante ('A' pour alphabétique, 'F' pour numérique,  
'D1' pour une date)

FORMEXT : format externe de la composante

PRECISION : précision de la composante

BUFFER : type de buffer (lié à TYPE) ('1' pour alphabétique, '7' pour  
numérique ou date)

DEPLACEMENT : déplacement de la composante par rapport au début du fichier

FORMINT : format interne de la composante

CLE : indicateur de clé ( = '1' si la composante est une clé, '0' sinon)

#### 6.3.2 La base de données descriptive

De la base de données descriptive, telle que définie dans (BDER), nous avons éliminé :

- des informations servant de documentation à l' utilisateur consultant la base de données descriptive (cfr. (6) et (7))
- des informations déjà présentes dans le descriptif existant (cfr. (8))
- des informations uniquement utilisées lors de l' application de consultation (cfr. (1) et (5))



Détaillons les différentes modifications a portées : (notons que les composantes soulignées en traits pointillés (-----) sont les composantes qui n' ont pas été implémentées)

1) APPLICATION (APPLIC,RELOGID,USERID,DESCRIPTION)

cette relation n' a pas été définie car uniquement utilisée lors de la consultation.

2) COMPOSANT (RELPHID,DATAID,NUMDATA,TITDATA,MESURE,POSSAI,DEPLACEMENT)

nous avons éliminé les composantes TITDATA, MESURE et POSSAI car elles ne sont utiles que dans le cadre de la consultation (informations), la composante DEPLACEMENT est reprise dans le descriptif des composantes (cfr. 6.3.1.2)

3) DATATYPE (DATAID,DOMID,PRIMAIRE)

la notion de domid ne nous sert plus à rien car le type, le format interne et le format externe sont définis au niveau du DATAID dans les descriptifs particuliers (cfr. 6.3.1.2). Il ne servirait donc à rien de regrouper toutes les composantes de même type, même format interne et externe.

4) KEYUNIT (KEYID,UNITID\_1,UNITID\_2,CARDINAL)

nous avons éliminé les composantes UNITID 1 et UNITID 2 car on peut les retrouver facilement à l' aide du descriptif des composantes.

5) PERMISSION (USERID,RELPHID,AGREG)

cette relation n' a pas été implémentée car elle est uniquement utilisée lors de la consultation.

6) RELLOG (RELOGID,TITLOG,DIMREF,MVT,NBMXVR,KEYID,READPASS)

nous avons supprimé, les composantes TITLOG et DIMREF car elles servent uniquement à renseigner l' utilisateur, ainsi que la composante READPASS qui ne sert que lors de la consultation.

7) RELPHY (RELPHID,TITPHY,ADRPHY,USERID,FCOL,WRPASS,RELOGID,DIVULG)

la composante TITPHY ne sert qu' à renseigner l' utilisateur de même que la composante FCOL, la composante DIVULG ne sert qu' à la consultation et ADRPHY est déjà reprise au niveau du descriptif général c' est pourquoi nous les avons supprimées.

8) STAND (DATAID,DOMID,FORMINT,FORMEXT)

la relation STAND n' a pas été implémentée pour des raisons expliquées en 3

9) UNITTYPE (UNITID,CARDINAL,DATAID)

la relation UNITTYPE a été supprimée pour des raisons expliquées en 4, de plus CARDINAL n' est utilisé que lors de la consultation.

10) COMDATA (DATAID,FORMULE,PROGR)

UTILISATEUR (USERID,PASSWORD,NOM)

VERSIONS (RELPHID,PREF,ETAT,CARDIN)

Ces relations n' ont subi aucune modification c' est à dire qu' elles ont été implémentées telles quelles.

#### 6.4 Création physique de la base de données

---

La création de la base de données comporte trois étapes :

- création des descriptifs des relations, composantes
- chargement des valeurs des composantes des relations
- opérations à effectuer en vue de rendre la base de données interrogeable.

##### 6.4.1 1ère étape : création des descriptifs des relations, composantes

---

Rappelons que le langage de programmation utilisé est APL. Les principales variables APL utilisées sont :

- un tableau REL qui est le descriptif général des relations
- des tableaux descriptifs des composantes de chaque relation physique désignés par le caractère 'C' suivi du nom de la relation qu' ils représentent.

Exemple : La relation physique 1.1 intitulée *identifiant de l' établissement* qui s' écrit IDET1 (ET,NOM,ADR,DATECR,EP,SECTACT1,SECTACT2,SECTACT3) sera représentée dans le descriptif général par :

IDET1 8 FIDET1 DBER A1 144

où IDET1 : nom mnémonique de la relation physique

8 : nombre de composantes de la relation

FIDET1 DBER A1 : nom du fichier contenant cette relation

144 : longueur d' une occurrence de ce fichier (tuple)

Cette même relation physique sera représentée par le descriptif particulier CIDET1 dont la forme est la suivante :



## CIDET1

```

1 ET A 15 0 1 1 11 1
2 NOM A 44 0 1 12 40 0
3 ADR A 64 0 1 52 60 0
4 DATECR D1 14 0 7 112 10 0
5 EP A 14 0 1 122 11 0
6 SECTACT1 A 6 0 1 133 4 0
7 SECTACT2 A 6 0 1 137 4 0
8 SECTACT3 A 6 0 1 141 4 0

```

#### 6.4.2 2ème étape : chargement des valeurs

Nous pourrions parler de base de données fictive car nous n' avons crée qu' un nombre limité d' occurrences par relation pour toutes les relations du subset : par exemple 5 entreprises et 6 établissements. La création de plusieurs milliers d' occurrences ne pose cependant pas de problèmes particuliers. Toutefois le temps de réponse lors de la mise à jour ou de la consultation de la base de données en sera considérablement augmenté (cfr. 6.5.7).

La création des valeurs des composantes de chaque relation s' effectue, dans notre cas, interactivement. L' utilisateur possède le choix de créer des valeurs pour toutes les composantes d' une relation ou seulement pour une partie de celles-ci. L' utilisateur fournit les valeurs des composantes à créer, tuple par tuple, à la fonction APL (CREREL) qui se charge, à la fin de la création, de stocker la relation créée dans le fichier correspondant à cette relation physique.

Grâce aux informations contenues dans les descriptifs, le programme de création effectuera les vérifications suivantes :

- le relation à créer doit préalablement être définie dans le tableau REL (descriptif général) et posséder son descriptif particulier.
- l' utilisateur doit créer des composantes appartenant à la relation à construire (descriptif particulier)
- contrôle du type et du format de chaque composante.

Exemple : Créons des valeurs pour les composantes de la relation 3.3 intitulée *dettes au Ministère des Finances pour impôt des sociétés* de la forme

DIS (EP,AN,MOISRI,MONTANT)

où EP : identifiant de l' entreprise

AN : dimension de référence

MOISRI : nombre de mois de retard de paiements

MONATANT : montant de la dette

Procédure de création

Notons que les caractères de type *abcd ABCD* sont ceux fournis par l'utilisateur.

*CREREL*

relations définies : IDET1, IDET2, IDEP, DPP, DONSS, DIS, etc...

relation à construire ?

*DIS*

composantes de la relation DIS

EP AN MOISRI MONTANT

liste des composantes à créer ?

*EP AN MOISRI MONTANT*

liste des composantes qui seront créées :

EP

AN

MOISRI

MONTANT

ok ? oui/non

*oui*

pour terminer la création entrez le caractère '\*'

tuple ?

EP :

*00412345601*

AN :

*77*

MOISRI :

*1*

MONTANT :

*100000*

tuple ?

EP :

\*

relation DIS chargée

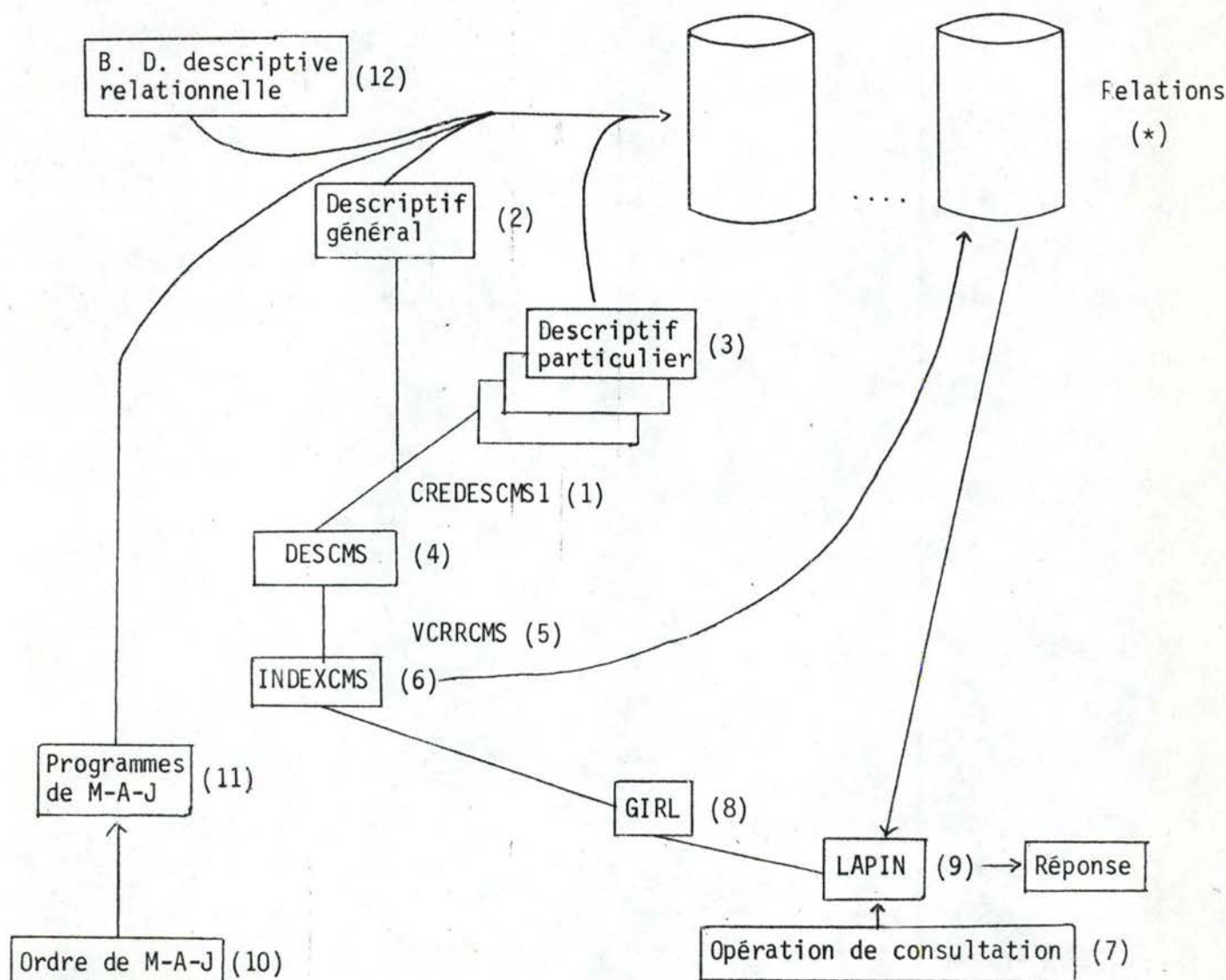


### 6.4.3 3ème étape : opérations à effectuer en vue de rendre la base de données interrogeable

Une fonction (CREDESCMS1) (1) prend le descriptif général des relations (REL) (2) ainsi que les descriptifs des composantes (3) et crée un fichier (DESCMS) (4) contenant la description externe de chaque relation physique (\*). Un programme (VCRRCMS) (5) crée un fichier (INDEXCMS) (6) contenant la description interne de chaque relation physique. Ce fichier sera exploité lors de la consultation (7) de la base de données (\*) par un interface (GIRL) (8) qui connaîtra ainsi l'existence et le format des relations et pourra répondre aux questions posées par l'utilisateur grâce au langage LAPIN (9).

Pour la mise à jour (10) les programmes de mise à jour (11) utilisent la base de données descriptive (12), le descriptif général (2) ainsi que les descriptifs particuliers pour mettre à jour la base de données.

Notons que l'utilisateur trouvera une description détaillée de DESCMS1 dans (DESCM) et de GIRL dans (GIRL).



- figure 6 -

## 6.5 Application de consultation réalisée grâce au 'LAPIN'

La consultation de la base de données a été effectivement réalisée par l'intermédiaire d'un langage d'interrogation existant appelé 'LAPIN' (LAngage Pour Interrogation). Il nous était, en effet, impossible, faute de temps, de définir notre propre langage d'interrogation pourtant indispensable à une base de données aussi particulière que la base de données économiques régionale relationnelle proposée par (BDER), si on veut réaliser les meilleures performances possibles. C'est pourquoi nous avons utilisé LAPIN. Nous décrivons celui-ci brièvement et présentons ses opérations de consultation en insistant sur celles qui nous intéressent particulièrement dans notre système avant de faire des suggestions en vue de l'améliorer.

### 6.5.1 Présentation du langage LAPIN

Le LAPIN est un langage d'interrogation interactif de bases de données qui n'est pas incorporé dans un langage hôte. Il possède des extensions permettant des questions quasi-naturelles. Cette facilité offre la possibilité de travailler avec un langage facile à apprendre et à manipuler même à des utilisateurs non initiés. Les opérateurs formels restent évidemment à la disposition des spécialistes qui veulent poser des questions avec un maximum de précision. Après avoir reconnu l'utilisateur, le langage met à sa disposition une gamme d'opérations facilement compréhensibles.

Nous allons examiner les opérations suivantes de LAPIN :

- le listage d'une relation
- la projection d'une relation sur une liste de composantes
- sélection sur filtre
- jointure de deux relations sur une liste de composantes
- fonctions horizontales et verticales
- combinaison des opérations

Le lecteur trouvera les algorithmes de ces opérations en 5.4.1

### 6.5.2 Listage d'une relation grâce au LAPIN

En consultant une relation l'utilisateur obtient toutes les données présentes dans la base de données pour cette relation.

#### 6.5.2.1 Format général en LAPIN : l'utilisateur indique le nom de la relation



## 6.5.2.2 Exemples

EX 1 : Question : CAPASSIF

Réponse : CAPASSIF

| EP          | AN | VALPAS   |
|-------------|----|----------|
| 00412345601 | 76 | 18931000 |
| 00423456702 | 76 | 5907000  |
| 00434567803 | 76 | 14860000 |
| 00445678904 | 76 | 7000000  |
| 00456789005 | 76 | 59300000 |
| 00412345601 | 77 | 15100000 |
| 00423456702 | 77 | 5000000  |
| 00434567803 | 77 | 18000000 |
| 00445678904 | 77 | 8000000  |
| 00456789005 | 77 | 70000000 |

La relation CAPASSIF n° 5 intitulée *Capital passif* a la description suivante :

EP : identifiant de l'entreprise (composante primaire)

AN : année (composante primaire temporelle)

VALPAS : valeur totale des postes de passif

La relation contient des données de 2 versions ( 76 et 77)

EX 2 : Question : IDEP

Réponse : IDEP

| EP          | ETHR        |
|-------------|-------------|
| 00412345601 |             |
| 00423456702 |             |
| 00434567803 | 11434567803 |
| 00445678904 |             |
| 00456789005 |             |

ADR

|                       |      |      |           |
|-----------------------|------|------|-----------|
| CHAUSSÉE DE WATERLOO  | 20   | 1080 | BRUXELLES |
| PARC INDUSTRIEL       | 1304 |      | BIERGES   |
| CHAUSSÉE DE BRUXELLES | 14   | 1320 | GENVAL    |
| CHAUSSÉE DE BOITSFORT | 29   | 1300 | WAVRE     |
| BOULEVARD DU MOULIN   | 136  | 1400 | NIVELLES  |

NOM

DATECR

VANDEMOORTELE CONSERVES  
 RAFFINERIE TIRLEMONTTOISE  
 ZWAN CONSERVES  
 IMELCO LAITERIE  
 SCIERIES REUNIES

01/01/1957  
 01/01/1936  
 01/06/1947  
 01/09/1922  
 01/01/1962

FJUR SECTACT1 SECTACT2 SECTACT3

|     |      |      |      |
|-----|------|------|------|
| 060 | 0062 | 0068 | 0070 |
| 120 | 0066 | 0000 | 0000 |
| 064 | 0060 | 0000 | 0000 |
| 120 | 0127 | 0000 | 0000 |
| 060 | 0120 | 0000 | 0000 |

La relation IDEP n° 2 intitulée *identifiant de l' entreprise* de forme :

EP : identifiant de l' entreprise (composante primaire)

ETHR : identifiant des établissements (ET) hors région

ADR : adresse EP (siège social)

NOM : nom ou raison sociale de EP

DATECR : date création EP

FJUR : forme juridique EP

SECTACTn : secteur d' activité n de EP

La relation IDEP contient donc 5 entreprises.

EX 3 : Question : *LICENC*

—— Réponse : *LICENC*

| EP          | AN | NBLIC |
|-------------|----|-------|
| 00412345601 | 76 | 10    |

Une seule entreprise à licencié (10 personnes) et c' était en l' année 1976

La relation LICENC n° 4 intitulée *licenciement* et dont la forme est :

EP : identifiant de l' entreprise (composante primaire)

AN : année (composante primaire temporelle)

NBLIC : nombre de personnes licenciées collectivement au cours  
de l' année

EX 4 : Question : *CAPASS*

—— Réponse : La relation CAPASS n' existe pas

En effet, nous n' avons pas défini de relation nommée CAPASS et en  
conséquence LAPIN imprime un message d' erreur (cfr. annexe 1)

### 6.5.3 Projection d' une relation sur une liste de composantes grâce au LAPIN

---

#### 6.5.3.1 Format général

---

L % R

où L : liste de composantes de R

R : nom d' une relation

% : opérateur de projection

#### 6.5.3.2 Exemples

---



EX 5 : Question : *AN % CAPASSIF*

on recherche toutes les années pour lesquelles on a des données dans CAPASSIF

Réponse : CAPASSIF

AN

76

77

La relation CAPASSIF est projetée sur sa composante AN. Cette opération élimine les duplications.

EX 6 : Question : *EP,AN % CAPASSIF*

on recherche tous les couples (entreprise,année) pour lesquels on a des données dans CAPASSIF

Réponse : CAPASSIF

EP

AN

|             |    |
|-------------|----|
| 00412345601 | 76 |
| 00412345601 | 77 |
| 00423456702 | 76 |
| 00423456702 | 77 |
| 00434567803 | 76 |
| 00434567803 | 77 |
| 00445678904 | 76 |
| 00445678904 | 77 |
| 00456789005 | 76 |
| 00456789005 | 77 |

Le résultat de la projection a été trié sur les valeurs de la première composante primaire (EP)

EX 7 : Question : *AN,EP % CAPASSIF*

on recherche tous les couples (année,entreprise) pour lesquels on a des données dans CAPASSIF

Réponse : CAPASSIF

AN

EP

|    |             |
|----|-------------|
| 76 | 00412345601 |
| 77 | 00412345601 |
| 76 | 00423456702 |
| 77 | 00423456702 |
| 76 | 00434567803 |
| 77 | 00434567803 |
| 76 | 00445678904 |
| 77 | 00445678904 |
| 76 | 00456789005 |
| 77 | 00456789005 |

Le résultat est trié sur les valeurs de la première composante primaire (EP) comme pour l'exemple précédent.

Notons que les composantes occupent le même rang dans la réponse et dans la question.

EX 8 : Question : *EP,AN,VALPAS % CAPASSIF*

on recherche tous les triplets (entreprise,année,valeur)  
pour lesquels on a des données dans CAPASSIF

Réponse : CAPASSIF

| EP          | AN | VALPAS   |
|-------------|----|----------|
| 00412345601 | 76 | 18931000 |
| 00412345601 | 77 | 15100000 |
| 00423456702 | 76 | 5907000  |
| 00423456702 | 77 | 5000000  |
| 00434567803 | 76 | 14860000 |
| 00434567803 | 77 | 18000000 |
| 00445678904 | 76 | 7000000  |
| 00445678904 | 77 | 8000000  |
| 00456789005 | 76 | 59300000 |
| 00456789005 | 77 | 70000000 |

Si la liste des composantes contient toutes les composantes de la relation, le résultat de l'opération sera identique à celui du listage de la relation (cfr. 6.5.2.2 ex. 1) si lors de la projection les composantes sont triées selon le rang qu'elles occupent dans la définition de la relation. CAPASSIF a été défini comme suit : CAPASSIF (EP,AN,VALPAS)

où EP : argument de tri primaire

AN : argument de tri secondaire

VALPAS : argument de tri tertiaire

#### 6.5.4 Sélection sur filtre grâce au LAPIN

##### 6.5.4.1 Format général

R ; { E }

où R : nom de relation

{ E } : ensemble d'expressions du type C op V

où C est le nom d'une composante

V est une valeur

op est un opérateur de comparaison

##### 6.5.4.2 Exemples

EX 9 : Question : *CAPASSIF ; AN = '77'*

on recherche les tuples de CAPASSIF pour l'année 77



Réponse : CAPASSIF

| EP          | AN | VALPAS   |
|-------------|----|----------|
| 00412345601 | 77 | 15100000 |
| 00423456702 | 77 | 5000000  |
| 00434567803 | 77 | 18000000 |
| 00445678904 | 77 | 8000000  |
| 00456789005 | 77 | 70000000 |

Le résultat n' est autre que la consultation d' une version de la relation CAPASSIF

EX 10 : Question : *CAPASSIF ; VALPASS > 0*

Réponse : column named VALPASS not found in relation CAPASSIF

Il y a, en effet, une erreur : on n' a pas défini de composante VALPASS pour la relation CAPASSIF

EX 11 : Question : *CAPASSIF ; VALPAS > '0'*

Réponse : error 612 condition conversion

Le message d' erreur spécifie que la valeur n' est pas du même type que la composante correspondante dans le filtre. En effet, VALPAS est de type numérique dans le descriptif des composantes (cfr. annexe 2 ).

EX 12 : Question : *CAPASSIF ; VALPAS < 10000000*

on recherche tous les tuples de CAPASSIF pour lesquels la valeur du capital passif est < 10000000

Réponse : CAPASSIF

| EP          | AN | VALPAS  |
|-------------|----|---------|
| 00423456702 | 76 | 5907000 |
| 00445678904 | 76 | 7000000 |
| 00423456702 | 77 | 5000000 |
| 00445678904 | 77 | 8000000 |

Il s' agit d' une sélection sur un filtre formé d' une expression simple. L' exemple suivant montre un filtre formé de 2 expressions (qui doivent être vraies toutes les deux)

EX 13 : Question : *CAPASSIF ; AN = '76' et VALPAS < 10000000*

on recherche tous les tuples de CAPASSIF pour lesquels la valeur de VALPAS < 10000000 pour l' année 76

Réponse : CAPASSIF

| EP          | AN | VALPAS  |
|-------------|----|---------|
| 00423456702 | 76 | 5907000 |
| 00445678904 | 76 | 7000000 |

## 6.5.5 Jointure de deux relations par LAPIN

### 6.5.5.1 Format général

$R1 ; C1 = C2 \% R2$

où  $R1, R2$  : noms de relation

$C1$  : composante de  $R1$  intervenant dans la jointure

$C2$  : composante de  $R2$  intervenant dans la jointure

Le résultat se présentera sous la forme suivante :

|      |           |              |                        |
|------|-----------|--------------|------------------------|
| $R1$ |           | $R2$         | (noms des relations)   |
| $C1$ | $C2$ .... | $Ci...$ $C2$ | (noms des composantes) |
| ...  | ...       | ...          | (valeurs)              |

On peut donner en LAPIN une définition de la jointure sur deux composantes équivalente à celle donnée en 5.4.1.2.3 :

$R1 ; C1 = C2, (\text{autres composantes de } R2) \% R2$

Il reste malgré tout une restriction : on ne peut pas définir en LAPIN la jointure de 2 relations sur une liste de composantes; LAPIN offre, cependant, la possibilité de définir une jointure sur une composante qui regroupe plusieurs composantes adjacentes. (cfr. 6.5.8)

EX 14 : Question :  $LICENC ; EP = EP \% IDEP$

on recherche toutes les données de la relation LICENC pour les entreprises pour lesquelles on a des données signalétiques.

Réponse : LICENC IDEP

|    |    |       |    |
|----|----|-------|----|
| EP | AN | NBLIC | EP |
|----|----|-------|----|

|             |    |    |             |
|-------------|----|----|-------------|
| 00412345601 | 76 | 10 | 00412345601 |
|-------------|----|----|-------------|

Notons que le résultat contient deux fois les valeurs de la composante sur laquelle on a défini la jointure; on n'y trouve aucune valeur pour les composantes de la 2ème relation n'intervenant pas directement dans la jointure.

EX 15 : Question :  $LICENC ; EP = EP, ADR \% IDEP$

on recherche toutes les données de la relation LICENC pour les entreprises pour lesquelles on a des données signalétiques et dont on veut aussi l'adresse



Réponse : LICENC IDEP

| EP          | AN | NBLIC | EP          | ADR                                    |
|-------------|----|-------|-------------|----------------------------------------|
| 00412345601 | 76 | 10    | 00412345601 | CHAUSSÉE DE WATERLOO 20 1080 BRUXELLES |

Si l' utilisateur désire imprimer les valeurs des composantes de la 2ème relation, sur lesquelles la jointure n' a pas été définie, il doit le demander explicitement en projetant la relation sur celles-ci.

#### 6.5.6 Fonctions horizontales et verticales de LAPIN

L' adjonction de nouvelles composantes à une relation c' est à dire la réalisation des fonctions horizontales n' existe pas dans LAPIN. Il est, cependant possible, d' effectuer des opérations verticales sur une composante d' une relation, par exemple : la somme, le nombre minimum, le nombre maximum et la moyenne de valeurs d' une composante, et d' en demander des valeurs intermédiaires.

EX 16 : Question : *SOMME des VALPAS de CAPASSIF par AN*

Réponse : CAPASSIF

| AN | SOMME    |
|----|----------|
| 76 | 05998000 |
| 77 | 16100000 |

On a calculé pour chaque année la somme des valeurs de CAPASSIF. Nous remarquons qu' il y a overflow, en effet, la somme réelle pour l' année 76 est de 105998000 et pour l' année 77 de 116100000. (8 chiffres édités seulement !)

EX 17 : Question : *SOMME des VALPAS de CAPASSIF par EP*

Réponse : CAPASSIF

| EP          | SOMME    |
|-------------|----------|
| 00412345601 | 34031000 |
| 00423456702 | 10907000 |
| 00434567803 | 32860000 |
| 00445678904 | 15000000 |
| 00456789005 | 29300000 |

On a calculé pour chaque entreprise la somme des valeurs du capital passif pour les versions (années) présentes dans la relation.

### 6.5.7 Temps de réponse de LAPIN

Les exemples qui vont suivre nous ont amené à la conclusion suivante : si le nombre de tuples d' une relation devient élevé (5000 occurrences par exemple) le temps de réponse reste très acceptable et n' augmente certainement pas proportionnellement au nombre d' éléments.

EX 18 : Question : *CAPACTIF*

Réponse : CAPACTIF

| EP          | AN | VALACT   |
|-------------|----|----------|
| 00412345601 | 76 | 19000000 |
| 00423456702 | 76 | 6000000  |
| 00434567803 | 76 | 18000000 |
| 00445678904 | 76 | 7000000  |
| 00456789005 | 76 | 70000000 |
| 00412345601 | 77 | 19000000 |
| 00423456702 | 77 | 6000000  |
| 00434567803 | 77 | 18000000 |
| 00445678904 | 77 | 7000000  |
| 00456789005 | 77 | 72000000 |
| 00412345601 | 78 | 72000000 |

CAPACTIF est une relation regroupant :

EP : n° entreprise (composante primaire non temporelle)

AN : année (composante primaire temporelle)

VALACT : valeur du capital actif (composante non primaire)

Le temps de réponse peut être mesuré par le nombre de caractères '\*' qui précèdent la réponse (ici aucun) ce caractère est une unité de mesure du temps d' attente.

EX 19 : Afin de tester le temps de réponse pour un grand nombre d' occurrences nous avons ajouté des tuples à la relation CAPASSIF

Question : *NOMBRE de CAPASSIF*

Réponse : \*\*\*\*\* 5000

Il y a donc 5000 tuples dans la relation CAPASSIF. Le temps d' attente pour la réponse est nettement plus long mais reste acceptable.

EX 20 : Question : *NOMBRE de CAPASSIF ; AN = '78'*

Réponse : \*\*\*\*\* 1

Il n' y a qu' un tuple de la relation CAPASSIF contenant des valeurs pour l' année '78'.



## 6.5.8 Précautions pour réaliser des jointures sur une liste de composantes

en LAPIN

LAPIN permet de faire une jointure entre deux relations sur 'n' composantes adjacentes. Il est nécessaire, pour cela, de définir, dans le descriptif des composantes, une composante virtuelle qui regroupe les composantes adjacentes. Nous nous sommes limités à définir ces composantes virtuelles pour les composantes de la clé. Ainsi, par exemple, la relation CAPASSIF aura le descriptif suivant :

```

1 EA A 19 0 1 1 13 1
2 EP A 15 0 1 1 11 0
3 AN A 3 0 1 12 2 0
4 VALPAS F 12 0 7 14 10 0 (cfr. annexe 2)

```

EX 21 : Question : CAPASSIF ; EA = EA % CAPACTIF

on veut comparer les valeurs du capital passif et actif des entreprises pour chaque année.

Réponse :

\*\*\*\*\*CAPASSIF

CAPACTIF

| EA            | EP          | AN | VALPAS   | EA            |
|---------------|-------------|----|----------|---------------|
| 0041234560176 | 00412345601 | 76 | 18931000 | 0041234560176 |
| 0041234560177 | 00412345601 | 77 | 15100000 | 0041234560177 |
| 0041234560178 | 00412345601 | 78 | 20000000 | 0041234560178 |
| 0042345670276 | 00423456702 | 76 | 5907000  | 0042345670276 |
| 0042345670277 | 00423456702 | 77 | 5000000  | 0042345670277 |
| 0043456780376 | 00434567803 | 76 | 14860000 | 0043456780376 |
| 0043456780377 | 00434567803 | 77 | 18000000 | 0043456780377 |
| 0044567890476 | 00445678904 | 76 | 7000000  | 0044567890476 |
| 0044567890477 | 00445678904 | 77 | 8000000  | 0044567890477 |
| 0045678900576 | 00456789005 | 76 | 59300000 | 0045678900576 |
| 0045678900577 | 00456789005 | 77 | 70000000 | 0045678900577 |

| EP          | AN | VALACT   |
|-------------|----|----------|
| 00412345601 | 76 | 19000000 |
| 00412345601 | 77 | 19000000 |
| 00412345601 | 78 | 72000000 |
| 00423456702 | 76 | 6000000  |
| 00423456702 | 77 | 6000000  |
| 00434567803 | 76 | 18000000 |
| 00434567803 | 77 | 18000000 |
| 00445678904 | 76 | 7000000  |
| 00445678904 | 77 | 7000000  |
| 00456789005 | 76 | 70000000 |
| 00456789005 | 77 | 72000000 |

Le résultat est correct car il contient les 5 tuples de l'année 76 et 5 tuples pour l'année 77 et un tuple pour l'année 78 (ajouté pour CAPASSIF lors de la création des 5000 occurrences ) cfr. ex. 8, 18, 20.

### 6.5.9 Combinaisons des opérations de consultation grâce à LAPIN

L' utilisateur peut combiner les opérations définies dans LAPIN de 6.5.2 à 6.5.8;

EX 22 : Question : *DATECR % IDEP ; EP = EP % CAPASSIF ; AN = '78'*

on désire la date de création des entreprises pour lesquelles on a des informations dans CAPASSIF pour l' année 78

Réponse : (nous avons toujours 5000 occurrences dans CAPASSIF)

```
*****IDEP CAPASSIF
DATECR EP EP AN
01/01/1957 00412345601 00412345601 78
```

EX 23 : Question : *DATECR % IDEP ; EP = EP, VALPAS % CAPASSIF ; AN = '78'*

on désire la date de création, la valeur du capital passif des entreprises pour lesquelles on a des informations dans CAPASSIF pour l' année 78

Réponse :

```
*****IDEP CAPASSIF
DATECR EP EP VALPAS AN
01/01/1957 00412345601 00412345601 20000000 78
```

Notons que si une sélection est incluse dans une combinaison d' opérations, la réponse ne laissera pas apparaître l' entièreté du tuple. Pour ce faire, il faut en plus projeter la relation sur la liste des composantes n' intervenant ni dans la sélection ni dans la jointure.

### 6.5.10 Suggestions en vue d' améliorer le 'LAPIN' (1)

En vue d' améliorer le 'LAPIN' nous faisons quelques propositions dont une partie pourrait mieux adapter LAPIN au problème spécifique de (BDER).

Nous suggérons les modifications dans les domaines suivants :

- tri
- fonctions horizontales
- relations intermédiaires
- jointure sur des composantes non adjacentes
- élimination des duplications lors de la projection
- prise en charge de contraintes d' intégrité

(1) Ces propositions ont été discutées et soumises aux responsables du projet 'Langage d' interrogation LAPIN' au Centre Scientifique IBM France à Paris



### 6.5.10.1 Tris implicites ou explicites

---

On pourrait donner la possibilité à l'utilisateur de faire des tris à la consultation de la base de données (implicites ou explicites).

#### Exemples : tri implicite

En écrivant  $C1, C2 \% R1$  on obtiendrait un résultat trié sur  $C1$

En écrivant  $C2, C1 \% R1$  on obtiendrait un résultat trié sur  $C2$

où  $C1, C2$  sont des composantes de la relation  $R1$

Le résultat serait, ainsi, trié sur la première composante

#### tri explicite

En écrivant  $R1 (T) C1$   $R1$  serait triée sur  $C1$  ou

En écrivant  $R1 (T) C1, C2$   $R1$  serait triée avec comme argument primaire  $C1$  et comme argument secondaire  $C2$

où  $T$  est un opérateur de tri

### 6.5.10.2 Fonctions horizontales

---

Il serait utile de pouvoir faire des fonctions horizontales sur des composantes appartenant à une même relation ou appartenant à des relations différentes. Il serait, en effet, intéressant de gagner de la place (composante résultant de la fonction horizontale) car on n'ajoute la composante que lors de la consultation.

Soit  $R (E)$  et  $C = f (E)$

où  $R$  : nom de relation

$E$  : ensemble des composantes de  $R$

$C$  : composante n' appartenant pas à  $R$  mais est une fonction horizontale de  $E$

$f$  : fonction horizontale

Actuellement on doit définir  $R$  comme étant une relation ayant  $\text{Card}(E+1)$  composantes (perte de place).

Si on dispose de fonctions horizontales, en listant  $R$  on obtient non seulement les composantes  $E$  mais également (et sans avoir à le demander) la composante  $C$  résultant de la fonction horizontale.

### 6.5.10.3 Fonctions verticales

---

En ne demandant que le listage d' une relation, l' utilisateur peut désirer de voir lister, non seulement les composantes de la relation, mais également des zones résultats découlant du calcul de fonctions verticales.

### 6.5.10.4 Relations intermédiaires

---

Il serait utile de pouvoir créer, à partir d' une relation, une ou plusieurs relations 'intermédiaires'.

Prenons un exemple : définissons R1 comme suit :

relation R1 (par établissement, par produit, somme des ventes)

En demandant R1 on voudrait également obtenir R2 (par établiss., somme ventes) sur laquelle on pourrait, ensuite faire une projection, selection, etc....

### 6.5.10.5 Jointure sur des composantes non adjacentes

---

Il devrait être possible de faire des jointures sur n composantes non adjacentes ( $n > 1$ ) entre deux relations.

Prenons comme exemple : R1 (A1,B1,C1) et R2 (A2,B2,C2)

où R1,R2 : noms de relation

A1,B1,C1 : composantes de R1

A2,B2,C2 : composantes de R2

Il serait possible de faire :  $R1 ; (A1=A2) \text{ et } (C1=C2) \% R2$  ou

$R1 ; (A1,C1) = (A2,C2) \% R2$

### 6.5.10.6 Elimination des duplications

---

Lors de projections, l' utilisateur devrait pouvoir décider s' il désire ou non l' élimination des duplications.

### 6.5.10.7 Contraintes d' intégrité

---

Le 'LAPIN' qui se base sur le descriptif existant devrait être capable de prendre en charge d' autres contraintes d' intégrité



## 6.6 Application de mise à jour

L'application de mise à jour se base totalement sur les relations de la base de données descriptive présentée en 6.3.2 ainsi que sur le descriptif existant (cfr. 6.3.1). Nous nous limiterons à l'implémentation de la mise à jour ponctuelle de la base de données en nous basant sur les algorithmes décrits en 5.5.1.1. La mise à jour massive suit le même principe que la mise à jour ponctuelle en ce qui concerne la vérification des données. Dans le cadre de notre réalisation la mise à jour de la base de données implique deux commandes :

- SIGNON pour la reconnaissance de l'utilisateur
- MAJ pour la mise à jour ponctuelle de la base de données

Détaillons chacune d'entre elles :

### 6.6.1 Reconnaissance de l'utilisateur

Pour mettre à jour la base de données, l'utilisateur doit être connu du système comme un utilisateur autorisé. Cette commande aura la forme suivante :

SIGNON userid,password,applic

où userid : nom de l'utilisateur

password : mot de passe de l'utilisateur

applic : application que l'utilisateur désire effectuer avec les données qu'il tire de la consultation. Applic DOIT donc être = Ø pour la mise à jour

### Procédure de reconnaissance

*SIGNON*

entrez userid,password,applic

U1,NAMUR1

\*\*\*userid ok

\*\*\*password ok

\*\*\*applic ok

### Contrôles effectués :

a) (userid) ∈ (USERID) UTILISATEUR

b) (userid,password) ∈ (USERID,PASSWORD) UTILISATEUR

c) applic = Ø pour la mise à jour

### 6.6.2 Mise à jour ponctuelle de la base de données

Le gestionnaire de la base de données est le seul à pouvoir effectuer la mise à jour ponctuelle des données afin d'assurer la cohérence de celles-ci. Cette mise à jour consiste à mettre à jour une ou plusieurs composantes d'une relation physique soit en modifiant la valeur de composantes, soit en créant de nouveaux tuples, soit en supprimant des données.

Décrivons les informations que le gestionnaire de la base de données doit fournir pour mettre à jour celle-ci, ainsi que les contrôles effectués sur ces informations grâce à divers exemples :

- mise à jour du noyau (modification de valeurs)
- mise à jour du noyau (création de tuple)
- mise à jour de l'entrepôt (création de tuple)
- mise à jour de l'entrepôt (donnant lieu à un calcul)

Le format général de la commande est le suivant :

MAJ nom de relation, mot de passe en écriture de la relation

#### 6.6.2.1 Informations à fournir par le gestionnaire

- a) le nom de la relation physique à mettre à jour ainsi que son writepass. Il est rappelé que l'on ne peut pas mettre à jour une relation contenant des données calculées.
- b) le nom des composantes à mettre à jour. Ce sont ceux définis dans le descriptif des composantes de chaque relation (cfr. annexe 2)
- c) la valeur de la clé (composante utilisée pour la recherche). Nous imposons à l'utilisateur d'accéder à un tuple uniquement par la clé : identifiant unique de chaque tuple. Nous avons vu que la dimension de référence est incluse dans la clé de chaque tuple (excepté pour les données de caractéristique de mouvement 'existant'). La base de données descriptive connaît ainsi automatiquement la dimension de référence du tuple mis à jour; l'utilisateur ne doit pas la fournir séparément. Le type d'opération que le gestionnaire désire effectuer est déterminé de la manière suivante : il s'agit, de l'ajoute d'un tuple si la valeur de la clé fournie n'existe pas, et de la modification ou de la suppression d'un tuple si la valeur de la clé fournie existe. (Pour la suppression il faut en plus positionner un indicateur de suppression).



d) la valeur des modifications

Après avoir sélectionné le tuple correspondant à la clé, la fonction de mise à jour indique, composante par composante, la valeur courante (qui peut être indéfinie) et sollicite la nouvelle valeur ou la modification.

6.6.2.2 Contrôles effectués lors de la mise à jour

- a) la relation que fournit l' utilisateur doit exister dans le descriptif général, doit appartenir à la B.D.D. (  $(relphid \in (RELPHID) RELPHY)$  ) et doit être non calculée (  $(relphid, '0') \in (RELPHID, CALCULEE) RELPHY$  ).
- b) le mot de passe en écriture de la relation physique doit correspondre à cette relation (  $(relphid, wrpass) \in (RELPHID, WRPASS) RELPHY$  )  
Pour les cas a) et b) en cas d' échec le gestionnaire doit à nouveau entrer le nom de la relation et son mot de passe. Cette opération peut être répétée au maximum deux fois après quoi l' utilisateur est éjecté du système.
- c) les composantes que l' utilisateur met à jour doivent exister dans le descriptif des composantes de la relation ainsi que dans la B.D.D. (  $(relphid, dataid) \in (RELPHID, DATAID) COMPOSANT$  ). Pour ce faire, il a le loisir de vérifier la liste des composantes de la relation et doit donner son accord explicite pour la continuation de la procédure. (cfr. 6.6.2.4 et suivantes).
- d) si la valeur courante est définie pour une composante et si l' utilisateur répond par (5) (cfr. (5) 6.6.2.3, le message '? valeur' s' imprime et il doit utiliser une des autres possibilités ( (1), (2), (3) ou (4) de 6.6.2.3).
- e) Pour les données de type numérique, toute réponse donnant une valeur non numérique donne lieu à un message d' erreur et à une nouvelle sollicitation.
- f) Si la chaîne de caractères fournie pour une composantes est trop longue, elle est tronquée et on passe à la composante suivante.

6.6.2.3 Types de modifications

Nous proposons quelques types de modification qui se retrouvent dans les exemples qui suivent. Prenons comme exemple la réponse au message :

VACHAT : 15200

où VACHAT : nom d' une composante de type numérique  
15200 : la valeur courante

les réponses suivantes pourraient survenir :

- 1) [RC] retour de chariot : pas de modification, on passe à la composante suivante
- 2) R/VALEUR : exemple : R/15300 : remplacement de l' ancienne valeur par 15300
- 3) R/[RC] : suppression de l' ancienne valeur
- 4) C/15200/150200/ : correction d' une chaîne de caractères : la chaîne 15200 sera remplacée par la chaîne 150200
- 5) VALEUR : si la valeur courante n' était pas définie, l' utilisateur peut, dans ce cas, créer la valeur de la composante
- 6) \* : pour terminer la mise à jour

#### 6.6.2.4 Exemple de procédure de mise à jour du noyau (modification)

-----

Cet exemple montre une mise à jour du noyau qui a pour effet de modifier la valeur de la date de début d' exercice pour un établissement.

MAJ (\*)

entrez relation,writepass

IDET2,EU1

relation ok \*\*\*

wypass ok \*\*\*

1 la relation à mettre à jour est : IDET2

composantes de la relation : IDET2

ET DATEDEX

liste des composantes à mettre à jour ?

2 DATEDEX

composantes qui seront modifiées :

DATEDEX

3 ok ? yes/no (y/n)

y

4 la composante utilisée pour la recherche sera 'ET'

ET signifie : établissement (11)

pour terminer la mise à jour entrez '\*'

argument de recherche : ET =

5 .....

00412345601

(\*) les petits chiffres à gauche des instructions renvoient à des commentaires se trouvant à la page suivante



IDET2 ?

DATEDEX : 01/01/1957

6 C/57/56

fichier FIDET2 DBER A1 mis à jour  
relation : IDET2 mise à jour  
pas de calcul à faire pour DATEDEX  
argument de recherche : ET =

.....

\*

temps de mise à jour :

0 H 0 M 48 S 754

encore des données à mettre à jour

yes/no (y/n) ?

N

#### REMARQUES

- comme on peut le voir dans les différents exemples de mise à jour, nous donnons le libellé en clair de chaque composante (temporelle ou non) de la clé primaire ainsi que la longueur de celles-ci afin de mieux renseigner l'utilisateur sur la clé qu'il aura à fournir.
- si une composante mise à jour est 'primaire' (est utilisée dans un calcul) l'utilisateur en est averti et il peut suivre le déroulement de celui-ci : il reçoit des messages du type 'pas de calcul à faire pour telle composante' ou 'le numéro de la formule dans laquelle cette composante est utilisée' ou encore 'le résultat du calcul' et la relation concernée etc...  
(cfr. exemples de mise à jour de l'entrepôt)

#### Commentaires

- 1) la relation IDET2 (n° 1.2) est une relation du noyau; sa caractéristique de mouvement est 'existant'.
- 2) la composante à mettre à jour est DATEDEX qui appartient à la relation IDET2
- 3) par sécurité, le système demande à l'utilisateur de confirmer les composantes à mettre à jour. En cas de réponse négative, l'utilisateur doit fournir à nouveau la liste des composantes à mettre à jour.

- 4) on oblige l' utilisateur d' accéder à la relation à mettre à jour par la clé primaire. On lui donne la longueur de la clé ainsi que le libellé en clair de celle-ci.
- 5) au cas où la clé contient un nombre important de caractères ce repère permet à l' utilisateur de vérifier s' il a fourni une clé de longueur exacte.
- 6) le gestionnaire de la base de données corrige une erreur : changer l' année 1957 en 1956.

#### 6.6.2.5 Exemple de procédure de mise à jour du noyau (création)

-----

MAJ

entrez relation,writepass

IDET2,EU1

relation ok \*\*\*

wrpass ok \*\*\*

la relation à mettre à jour est : IDET2

composantes de la relation : IDET2

ET DATEDEX

liste des composantes à mettre à jour ?

ET DATEDEX

composantes qui seront modifiées

ET DATEDEX

ok ? yes/no (y/n)

y

la composante utilisée pour la recherche sera 'ET'

ET signifie établissement (11)

pour terminer la mise à jour entrez '\*'

argument de recherche : ET =

.....

[RC]

1 pas trouvé

entrez '.CRE' pour création, retour de chariot pour continuer

2 .CRE

création, entrez la valeur de : ET =

.....

09876543210

3 cr = 09876543210 créé



IDET2 ?  
ET : 09876543210

?

[RC]

DATEDEX :

?

01/04/1979

fichier FIDET2 DBER A1 mis à jour  
relation : IDET2 mise à jour  
pas de calcul à faire pour DATEDEX

argument de recherche : ET =

.....

\*

temps de mise à jour :

0 H 1 M 29 S 385

encore des données à mettre à jour

yes/no (y/n) ?

N

### Commentaires

- 1) le gestionnaire de la base de données peut, s' il sait qu' il va devoir créer un tuple, ne pas entrer de valeur de la clé (retour de chariot), le système déclenchera alors une routine de création.
- 2) l' utilisateur doit manifester explicitement son intention de créer un tuple en entrant '.CRE'.
- 3) la valeur de la clé est créée il ne reste plus qu' à fournir les valeurs des autres composantes

### 6.6.2.6 Exemple de procédure de mise à jour de l' entrepôt (création)

---

MAJ

entrez relation,writepass

CAPP4,EU1

relation ok \*\*\*

wypass ok \*\*\*

la relation à mettre à jour est CAPP4

composantes de la relation CAPP4

EA MOYACT EVB

liste des composantes à mettre à jour ?

MOYACT

composantes qui seront modifiées :

MOYACT

ok ? yes/no (y/n)

Y

la composante utilisée pour la recherche sera 'EA'

EA signifie entreprise (11), année (2)

pour terminer la mise à jour entrez '\*'

argument de recherche EA =

.....

[RC]

pas trouvé

entrez '.CRE' pour la création, retour de chariot pour continuer

.CRE

création entrez la valeur de EA =

.....

0042345670279

1 signalétique FIDEP DBER A1 ok \*\*\*

cr = 0042345670279 créé

CAPP4 ?

MOYACT :

?

450

fichier FCAPP4 DBER A1 mis à jour

relation CAPP4 mise à jour

2 fichier FVERS DBER A1 mis à jour

pas de calcul à faire pour MOYACT

argument de recherche EA =

.....

\*

temps de mise à jour :

0 H 3 M 26 S 623

encore des données à mettre à jour

yes/no (y/n) ?

N



## Commentaires

- 1) la clé contient une composante appartenant au noyau. Il y a donc lieu d'effectuer la contrainte de la présence obligatoire afin de vérifier (ici dans la relation IDEP *identifiant de l'entreprise*) si la valeur (numéro) de l'entreprise possède des données signalétiques.
- 2) la valeur de la composante temporelle de la clé (ANNEE) est 79, or dans la relation VERSIONS de la base de données descriptive il n'existe pas encore de données concernant la relation CAPP4 pour cette période de référence; il faut donc enregistrer dans VERSIONS cette nouvelle version des données.

### 6.6.2.7 Exemple de procédure de mise à jour de l'entrepôt (calcul à effectuer)

---

MAJ

entrez relation,writepass

RESEX2,EU1

relation ok \*\*\*

wrpass ok \*\*\*

la relation à mettre à jour est RESEX2

composantes de la relation : RESEX2

EA VCHA VREX VPP VFEX

liste des composantes à mettre à jour ?

1 VCHA VREX VPP VFEX

composantes qui seront modifiées :

VCHA VREX VPP VFEX

ok ? yes/no (y/n)

y

la composante utilisée pour la recherche sera 'EA'

EA signifie entreprise (11), année (2)

pour terminer la mise à jour entrez '\*'

argument de recherche EA =

.....

0045678900577

RESEX2 ?

VCHA : 80000000

?

C/8/7/

VREX : 1000000

?

C/1/4/

VPP : 2000000

?

R/1000000

VFEX : 400000

?

C/4/1/

fichier FRESEXP2 DBER A1 mis à jour  
relation RESEX2 mise à jour

- 2 égalité VCHA dans la formule PC13  
fichier FRATFIN1 DBER A1 mis à jour  
résultat du calcul PC13 : 2.12121212
- 3 égalité VCHA dans la formule PC14  
fichier FRATFIN1 DBER A1 mis à jour  
résultat du calcul PC14 : 0.22571428
- 4 égalité VCHA dans la formule PC17  
fichier FRATFIN3 DBER A1 mis à jour  
résultat du calcul PC17 : 0.03714285
- 5 égalité VREX dans la formule PC35
- 6 égalité VPP dans la formule PC35
- 7 égalité VPP dans la formule PC36  
fichier FRESEXP1 DBER A1 mis à jour  
résultat du calcul PC36 : 1000000
- 8 égalité VPP dans la formule PC16  
fichier FRATFIN2 DBER A1 mis à jour  
résultat du calcul PC16 : 0.25706940
- 9 pas de calcul à faire pour VFEX
- 10 mise à jour lors des calculs d' une donnée primaire : VAF  
égalité VAF dans la formule PC37  
fichier FRESEXP1 DBER A1 mis à jour  
résultat du calcul PC37 : 1000000
- 11 mise à jour lors des calculs d' une donnée primaire : VCAF  
égalité VCAF dans la formule PC10  
fichier FRATFIN1 DBER A1 mis à jour  
résultat du calcul PC10 : 2



argument de recherche EA =

.....

\*

temps de la mise à jour :

0 H 10 M 44 S 98

encore des des données à mettre à jour

yes/no (y/n) ?

N

### Commentaires

- 1) les numéros des données VCHA, VREX et VPP sont respectivement 423001, 423002 et 423004 nous verrons plus loin que ce sont des données 'primaires' c'est à dire qu'elles sont utilisées dans le calcul d'autres composantes.
- 2) la composante VCHA (423001) est utilisée dans la formule de calcul PC13. En effet, le calcul de la composante ROTCAP fait intervenir les composantes 423001 (VCHA) et 422002.
- 3) la composante VCHA (423001) est utilisée dans la formule de calcul PC14. En effet, le calcul de la composante MARGE BR fait intervenir les composantes 423007 et 423001 (VCHA).
- 4) la composante VCHA (423001) est utilisée dans la formule de calcul PC17. En effet, le calcul de la composante RCTA fait intervenir les composantes 312005, 311001 et 423001 (VCHA).
- 5) la composante VREX (423002) est utilisée dans la formule de calcul PC35. Malheureusement le système ne dispose pas de toutes les données pour effectuer le calcul (pas même période de référence).
- 6) la composante VPP (423004) intervient dans la formule de calcul PC35. On peut faire la même remarque que pour (5).
- 7) la composante VPP (423004) est utilisée dans la formule de calcul PC36. En effet, le calcul de la composante VAF fait intervenir les composantes 424004 et 423004 (VPP).
- 8) la composante VPP (423004) est utilisée dans la formule de calcul PC16. En effet, le calcul de la composante TAFIN fait intervenir les composantes 424004, 423004 (VPP) et 324007.
- 9) il n'y a pas de calcul à faire dans le cadre de la mise à jour de la composante VFEX car celle-ci n'intervient dans aucun calcul.

- 10) la composante VAF calculée lors de (7) est elle-même primaire et va donc entraîner le calcul d' une composante : VACF (PC37).
- 11) la composante VACF est calculée à partir des composantes 423013 (VAF cfr.10) et 424001. On va donc exécuter le programme de calcul PC10.

Pour toutes les formules de calcul citées ci-dessus nous renvoyons à l' annexe 3 page 2.



## CHAPITRE 7    EBAUCHE D' IMPLEMENTATION DE LA BASE DE DONNEES AU MOYEN

---

### DU SYSTEME R

---

Après avoir décrit, le cheminement parcouru pour réaliser l'implémentation de la base de données proposée par (BDER), ainsi que les problèmes d'implémentation rencontrés, intéressons-nous à présent à ce qui existe comme système permettant l'implémentation et la gestion d'une base de données relationnelle : Système R, INGRES, ... Nous avons pris l'un d'entre eux, le Système R, afin de voir s'il peut être intéressant de réaliser l'implémentation de la base de données grâce à un système existant.

Nous présentons une vue générale du Système R ainsi que du langage associé SEQUEL par lequel on peut réaliser les applications de consultation et de mise à jour.

Notons, que l'implémentation utilise le descriptif proposé par le Système R et la base de données descriptive réduite à un minimum. De plus, signalons que nous n'avons pas eu le Système R à notre disposition. Pour écrire ces notes nous nous sommes, en effet, uniquement basés sur (SEQ2), (CHLD), (SYSR1), (SYSR2) et (ASLR).

#### 7.1 Présentation du Système R

Le Système R est un système de gestion de bases de données expérimental implémenté au laboratoire de recherche IBM à San José en Californie. Ce système est destiné à permettre une définition facile de bases de données ainsi que de leurs applications. Il s'appuie sur un modèle relationnel qui assure une compréhension aisée et qui permet l'utilisation d'opérateurs puissants tels que ceux de projection, sélection, jointure.

Le Système R supporte un langage, appelé SEQUEL, destiné à définir, accéder et modifier les données des relations. Ce langage, très évolué, a été défini pour différentes raisons : il est très facile à apprendre et à manipuler, même pour des non-informaticiens, il permet également l'utilisation d'un 'optimizer' afin d'améliorer les performances : ce qui permet au système de minimiser les coûts, car le langage spécifie seulement ce que l'on désire obtenir et non le chemin pour y arriver.

Le langage SEQUEL assure un maximum d'indépendance entre les programmes d'application et les données c'est à dire que lors de réorganisation des

données, les programmes d'application ne sont pas à modifier. Les données peuvent être, non seulement accédées et mises à jour grâce au langage SEQUEL, mais également par des programmes écrits en COBOL ou PL/1 contenant des instructions SEQUEL. En effet, les langages COBOL et PL/1 sont des langages hôtes qui possèdent une extension et acceptent l'incorporation d'instructions du langage SEQUEL.

Le Système R possède un mécanisme de gestion d'autorisation d'accès aux données. On peut, par exemple, définir des 'vues' de relations dans lesquelles se trouvent seulement des données qu'un tel utilisateur a le droit de consulter. D'autres facilités sont permises telles que : la possibilité de restaurer la base de données afin de pouvoir conserver un état cohérent de celle-ci. Une opération de mise à jour, composée, par exemple, de plusieurs instructions SEQUEL, peut échouer en cours d'exécution. Le système est capable de revenir en arrière, d'oublier les modifications déjà effectuées pour cette opération de mise à jour afin d'avoir à tout moment une base de données cohérente. Ceci entraîne que les modifications ne seront effectives qu'en cas de bon aboutissement de l'exécution de cette opération appelée 'transaction'.

Le Système R prévoit, également, un protocole lorsque plusieurs utilisateurs désirent travailler en concurrence sur les mêmes données. Des techniques de blocage sont utilisées pour résoudre des problèmes de concurrence et de synchronisation.

## 7.2 Points intéressants du Système R

Nous relevons ci-dessous quelques points intéressants proposés par le Système R tant pour la définition que pour la manipulation ou le contrôle des données.

### 7.2.1 Dictionnaire de données

Le Système R propose la définition d'un dictionnaire de données. Il contient à la fois des descriptions internes de la base de données, utilisées par le Système R, ainsi que des descriptions externes des données, compréhensibles par les utilisateurs. Celles-ci sont stockées dans des tables pouvant être lues et modifiées grâce au langage SEQUEL.

Avant de créer une relation, le responsable de la base de données ajoute un élément dans la table contenant la description de cette relation.



## 7.2.2 Utilisation du langage SEQUEL

Le langage SEQUEL présente un ensemble de facilités d'interrogation, de manipulation, de définition et de contrôle de données que nous allons tenter de définir brièvement. La plupart de ces facilités serviront à implémenter les applications de consultation et de mise à jour.

### 7.2.2.1 Facilités d'interrogation

Nous présentons les clauses les plus fréquemment utilisées en SEQUEL et donnons pour chacune d'entre elles un exemple.

#### 7.2.2.1.1 La clause SELECT

La clause SELECT permet de lister, soit toutes les données d'une relation (cfr. ex. 1), soit les valeurs d'un ensemble de composantes d'une relation (cfr. ex. 2).

EX 1 : on désire connaître toutes les données d'une relation

```
SELECT *
 FROM nom de relation
```

Notons que cette commande équivaut à la commande LIST (cfr. 3.6.1.1)

EX 2 : on ne s'intéresse qu'aux données d'une liste de composantes pour une relation

```
SELECT L
 FROM nom de relation
```

où L : liste de composantes

Notons que cette commande équivaut à PROJ ( R (L) ) (cfr. 3.6.1.2)

Si dans l'exemple 2 nous avons voulu éliminer les duplications nous aurions écrit SELECT UNIQUE ... Toutefois, insistons sur le fait que l'élimination des duplications pour des relations ayant un nombre assez élevé d'occurrences est une opération lourde et coûteuse.

#### 7.2.2.1.2 La clause WHERE

La clause WHERE permet de sélectionner des données d'une relation à partir d'un filtre. Elle contient un ensemble de prédicats (filtre) qui permettent de comparer une liste de composantes d'une relation avec des constantes (cfr. ex. 3), des expressions (cfr. ex. 4) ou des composantes de cette relation (cfr. ex. 5)

EX 3 : On sélectionne les tuples d' une relation à partir d' un filtre

```
SELECT *
 FROM nom de relation
 WHERE {L op V }
```

où L : liste de composantes  
 V : constante ou expression  
 op : opérateur de comparaison  
 {L op V } : filtre

Notons que cette commande équivaut à la commande  $SEL ( R \{L \text{ op } V\} )$  (cfr. 3.6.1.3)

EX 4 : On sélectionne les tuples d' une relation à partir d' un filtre

```
SELECT *
 FROM nom de relation
 WHERE {L1 op L2}
```

où L1, L2 : liste de composantes de nom de relation  
 op : opérateur de comparaison  
 {L1 op L2} : filtre qui compare les valeurs des composantes occupant le même rang dans leur liste respective.

#### 7.2.2.1.3 La clause IN

La clause IN permet de joindre 2 relations sur une liste de composantes.

EX 5 : Les données des 2 relations sont ramenées en une seule relation si la valeur de L1 est égale à celle de L2.

```
SELECT *
 FROM nom de relation 1
 WHERE L1 IN
 SELECT L2
 FROM nom de relation 2
```

où L1, L2 : liste de composantes respectivement de nom de relation 1 et 2

Cette commande aurait encore pu s' écrire :

```
SELECT *
 FROM nom de relation 1, nom de relation 2
 WHERE nom de relation 1.L1 = nom de relation 2.L2
```

Notons que cette commande équivaut à  $JOIN (R1(L1) * R2(L2))$  (cfr. 3.6.1.4)



#### 7.2.2.1.4 La clause GROUP BY

La clause GROUP BY permet de trier une relation sur les valeurs d' une ou plusieurs composantes. Les données de la relation seront triées en ordre croissant ou décroissant selon les valeurs de la liste de composantes.

EX 6 : SELECT \*

FROM nom de relation

ORDER BY liste de composantes

EX 7 : La commande suivante permet, grâce à la clause GROUP BY, d' obtenir une vue synchronique d' une relation en la triant sur une liste de composantes avec comme argument primaire les composantes non temporelles de la clé et comme argument secondaire les composantes temporelles de la clé.

SELECT \*

FROM nom de relation

ORDER BY (composantes non temporelles de la clé, composantes temporelles de la clé)

#### 7.2.2.1.5 Fonctions verticales

Le langage SEQUEL met à la disposition des utilisateurs un certain nombre de fonctions verticales (cfr. ex 8) telles que :

- la moyenne : AVG
- la somme : SUM
- le nombre : COUNT
- le maximum : MAX
- le minimum : MIN

L' utilisateur aura également la possibilité de définir lui-même ses propres fonctions verticales.

EX 8 : SELECT fctv (a)

FROM nom de relation

où fctv : une des fonctions verticales ci-dessus

a : expression arithmétique définie sur des composantes de la relation

#### 7.2.2.1.6 Intersection, Union, Différence

L' utilisateur peut faire des intersections (INTERSECT), des unions (UNION) ou des différences (MINUS) sur des composantes de deux relations.

EX 9 : un utilisateur désire obtenir, pour des composantes définies de la même manière (même type,...) dans 2 relations, le listage des composantes de la première relation qui n' appartiennent pas à la seconde.

```
SELECT L1
 FROM nom de relation 1
MINUS
SELECT L2
 FROM nom de relation 2
```

où L1, L2 : liste de composantes respectivement de nom de relation 1 et 2

#### 7.2.2.2 Facilités de manipulation

Il existe quatre catégories de manipulations qui permettent à l' utilisateur de modifier des valeurs dans les relations de la base de données. Il s' agit de -l' insertion d' un tuple dans une relation (INSERT)

-la suppression de tuples d' une relation (DELETE)

-la modification des valeurs d' une liste de composantes d' un tuple existant (UPDATE)

-la création d' une nouvelle relation à partir de données se trouvant dans d' autres relations (ASSIGN)

Donnons un exemple pour deux d' entre elles :

EX 10 : on désire ajouter un tuple à une relation tel que les composantes de L ont des valeurs occupant le même rang dans v.

```
INSERT INTO nom de relation (L) < v >
```

où L : liste de composantes de la relation (nombre = n)

v : liste de n valeurs

Si on a déclaré un identifiant unique pour cette relation, il y aura vérification, lors de l' ajoute d' un tuple à cette relation, de la non-existence de la clé dans la relation.



EX 11 : On supprime tous les éléments d' une relation qui satisfont au filtre.

DELETE nom de relation

WHERE {L op V}

où  $L$  : composante de la relation

V : valeur d' une occurrence de la composante de la relation

op : opérateur de comparaison

$\{L \text{ op } V\} : \text{filtre}$

#### 7.2.2.3 Facilités de définition

L'utilisateur peut créer et supprimer des relations, définir des 'vues' de relations et spécifier certains liens entre deux relations.

#### 7.2.2.3.1 Création d' une nouvelle relation

Format général : CREATE TABLE nom de relation

description des composantes de cette relation

De plus, il est possible de définir des synonymes.

Notons que la suppression d' une relation se fera par la commande

DROP nom de relation

#### 7.2.2.3.2 Spécification d' aides d' accès : LINK

Un lien (LINK) est pour le langage SEQUEL un ensemble de pointeurs qui relient des tuples d'une relation avec ceux d'une autre relation selon un certain critère (attribut) : d'après les valeurs d'une composante de la relation par exemple.

EX 12 : CREATE LINK nom du lien

FROM nom de relation 1(nom de composante de la relation 1)

T0 nom de relation 2(nom de composante de la relation 2)

Un tuple de la relation 1 et un élément de la relation 2 auront un lien privilégié s'ils ont des valeurs identiques dans leur composante respective.

#### 7.2.2.3.3 Définition de 'vues' de relations

Une 'vue' est une relation regroupant des composantes d' autres relations.

Une 'vue' est créée grâce à la clause VIEW; elle est dynamique c' est à dire qu' elle répercute les modifications des données des relations qu' elle contient.

Le format général est le suivant : DEFINE VIEW nom de la 'vue'

AS expression permettant de retrouver  
les valeurs des composantes voulues

Elle est supprimée par la commande DROP VIEW nom de la 'vue'

#### 7.2.2.4 Facilités de contrôle

-----

Les facilités de contrôle des données permettent, notamment, une surveillance de l'accès des utilisateurs aux données et assurent l'intégrité de celles-ci.

##### 7.2.2.4.1 Permission d'accès

Lorsqu'un utilisateur crée une relation ou une 'vue', il devra définir les droits d'accès aux données de celles-ci en spécifiant les opérations que d'autres utilisateurs pourront effectuer sur les données de ces relations.

Le format général de la commande est le suivant :

EX 13 : GRANT opération ON nom de relation  
TO users

où opération : type d'opération définie (INSERT, DELETE,  
UPDATE, ...)

users : nom des utilisateurs ayant obtenu le droit d'accès

L'opération inverse consiste à supprimer le droit d'accès par la commande suivante :

REVOKE opération ON nom de relation  
FROM users

##### 7.2.2.4.2 Vérification automatique de prédicats

Afin d'assurer l'intégrité des données, le Système R grâce à SEQUEL, offre la possibilité de faire des assertions c'est à dire de définir des prédicats dont la valeur booléenne, après évaluation, sera contrôlée :

- si celle-ci est fausse l'opération sera rejetée
- si celle-ci est vraie l'opération sera acceptée.

EX 14 : on désire vérifier un prédicat pour toute opération effectuée sur  
une relation

ASSERT nom de l'assertion  
ON nom(s) de relation(s)  
< prédicat >



#### 7.2.2.4.3 Déclenchement automatique de l'exécution d'instructions

Une opération peut être automatiquement exécutée après une occurrence d' une action spécifiée telle la lecture, l' ajoute, la suppression ou la modification d' un tuple d' une relation.

Le format général de la commande est le suivant :

```
EX 15 : DEFINE TRIGGER nom du trigger
 ON opération OF nom de la relation
 { instructions }
```

Lors de l'exécution de l'opération sur la relation, le TRIGGER déclenchera l'exécution de la liste d'instructions prévues à cet effet.

L'opération inverse consiste à supprimer un TRIGGER par la commande suivante :

```
DROP TRIGGER nom du trigger
```

### 7.3 Création de la base de données

La création des relations de la base de données consiste à créer une table par relation. Le format général de la commande est le suivant :

```
CREATE TABLE nom de la relation
 nom de composante 1 type format
 nom de composante 2 type format

 nom de composante n type format
```

Lors de la création de la base de données, il est possible de définir des synonymes par la commande suivante :

DEFINE SYNONYM nom de synonyme AS nom de relation ou  
nom de composante etc ...

Lors de la création de la base de données il est également possible de définir des 'vues' (cfr. 7.2.2.3,3). Rappelons que la modification des données d'une relation définies dans une 'vue' se répercute dans cette 'vue'.

Le format général de la commande est le suivant :

DEFINE VIEW nom du verrou AS expression permettant de retrouver les valeurs des données qu' on veut voir figurer dans la 'vue'

## 7.4 Application consultation

A chaque unité de consultation nous aurions pu faire correspondre une 'vue' (cfr. 7.4.2.6) , malheureusement, notre documentation ne nous a pas permis de connaître la répercussion au niveau physique (stockage redondant des données,...). C'est pourquoi nous avons préféré définir un programme général valable pour toutes les relations logiques pour les opérations suivantes :

- listage d' une relation logique
- projection d' une relation sur une liste de composantes
- sélection sur filtre
- jointure de deux relations

Ayant fait ce choix, nous gardons les relations RELLOG et RELPHY de la base de données descriptive (définie au chapitre précédent) car il nous faut, par exemple, pouvoir retrouver toutes les relations physiques d' une relation logique. En ce qui concerne les autres données de la B.D.D., utilisées lors de l' application de consultation, elles sont reprises dans la table de description des composantes (cfr. 7.3).

Les exemples de programmes décrivant les opérations de consultation sont écrits en PL/1 et contiennent des instructions SEQUEL (ce sont celles précédées par le caractère '\*' ). Nous supposons dans la suite que l' utilisateur a obtenu le droit d' effectuer ces opérations de consultation de la base de données.

Avant de détailler les programmes, expliquons quelques principes qui nous permettent de regrouper les relations physiques d' une relation logique sans avoir à définir des 'vues'.

1) \* LET prphy BE

\*     SELECT relphid INTO rphy

\*             FROM RELPHY

\*             WHERE RELOGID = relation logique;

= clauses

\* OPEN prphy;

  I = 0;

  DO WHILE (code = ok);

\*     FETCH prphy;

      I = I + 1;

      X(I) = rphy;

  END;

\* CLOSE prphy;



Expliquons cette partie de programme :

- la déclaration < LET prphy BE { clauses } > associe la variable prphy avec un ensemble de clauses.
- OPEN prphy : prépare la fourniture de tuples correspondants à la variable prphy
- FETCH prphy : chaque exécution de cette instruction transfère un nouveau tuple dans la variable rphy (spécifiée dans les clauses) si les conditions liées à cette variable sont satisfaites.
- code : cette variable permet d' indiquer la fin de la lecture (code = ok)
- X(n) : le vecteur X contient les n relations physiques d' une relation logique
- CLOSE prphy : cette instruction indique qu' il ne pourra plus avoir de référence à la variable prphy

2) \* SELECT keyid, dimref

\* FROM RELLOG

\* WHERE RELOGID = nom de la relation logique

Cette opération permet de connaître la clé primaire de la relation logique.

- keyid = composantes non temporelles de la clé primaire
- dimref = composantes temporelles de la clé primaire

Ces valeurs serviront comme argument de 'jointure' des relations physiques.

#### 7.4.1 Listage d' une relation logique

Rappelons que le listage d' une relation logique (cfr. 5.4.1.1) consiste à présenter toutes les données d' une relation logique à l' utilisateur.

Le format général de la commande est le suivant :

LIST nom de la relation logique

L' algorithme ayant été décrit en 3.6.1.1 nous présentons sa réalisation en langage PL/1 et SEQUEL.

LIST : PROCEDURE;

déclarations

\* LET prphy BE

\*     SELECT relphid INTO rphy

\*     FROM RELPHY

\*     WHERE RELOGID = relation logique;

\* SELECT keyid, dimref

\*     FROM RELLOG

\*     WHERE RELOGID = relation logique;

\* OPEN prphy;

  I = 0;

  DO WHILE (code = ok);

\*    FETCH prphy;

    I = I + 1;

    X(I) = rphy;

  END;

\* CLOSE prphy;

  /\* on désire effectuer le listage de la relation logique \*/

  /\* I contient le nombre de relations physiques de la relation logique \*/

  /\* on joint les relations physiques de la relation logique sur la clé primaire

    (composantes non temporelles et composantes temporelles) dans une relation \*/

  SELECT I

    WHEN (1)

  /\* la relation logique contient 1 relation physique \*/

\* SELECT \*

\*     FROM X(1);

    WHEN (2)

  /\* la relation logique contient 2 relations physiques \*/

\* SELECT \*

\*     FROM X(1),X(2)

\*     WHERE X(1).keyid = X(2).keyid

\*     AND X(1).dimref = X(2).dimref;

    WHEN (3)

  /\* la relation logique contient 3 relations physiques \*/

\* SELECT \*

\*     FROM X(1),X(2),X(3)

\*     WHERE X(1).keyid = X(2).keyid = X(3).keyid

\*     AND X(1).dimref = X(2).dimref = X(3).dimref;



WHEN (4)

/\* la relation logique contient 4 relations physiques \*/

```
* SELECT *
* FROM X(1),X(2),X(3),X(4)
* WHERE X(1).keyid = X(2).keyid = X(3).keyid = X(4).keyid
* AND X(1).dimref = X(2).dimref = X(3).dimref = X(4).dimref;

 OTHERWISE
 PUT SKIP ("la relation logique ne peut pas contenir plus de 4 relations phys");

 END;
END LIST;
```

#### 7.4.2 Consultation sélective

La consultation sélective permet à l'utilisateur d'éditer la partie des données d'une relation logique qui l'intéresse. Cet ensemble de données ne correspond donc pas à une relation logique entière.

Nous énumérons la description, en langage PL/1 et SEQUEL, des trois opérations élémentaires d'interrogation de base de données : projection, sélection, jointure. Nous abordons, enfin, d'autres opérations d'ordre secondaire.

##### 7.4.2.1 Projection d'une relation sur une liste de composantes

---

Format général : PROJ (R (L))

où R : nom de relation logique

L : liste de composantes de R

La projection d'une relation sur une liste de composantes peut se réaliser grâce au programme suivant :

PROJ : PROCEDURE;

déclarations

```
* LET prphy BE
* SELECT relphid INTO rphy
* FROM RELPHY
* WHERE RELOGID = relation logique;
* SELECT keyid,dimref
* FROM RELLOG
* WHERE RELOGID = relation logique;
```

```

* OPEN prphy;
 I = 0;
 DO WHILE (code = ok);
* FETCH prphy;
 I = I + 1;
 X(I) = rphy;
 END;
* CLOSE prphy;

/* on joint les relation physiques de la relation logique et on édite la liste
des composantes sur laquelle la relation est projetée */

SELECT I
 WHEN (1)

/* la relation logique contient une relation physique */
* SELECT L
* FROM X(1);
 WHEN (2)

/* la relation logique contient 2 relations physiques */
* SELECT L
* FROM X(1),X(2)
* WHERE X(1).keyid = X(2).keyid
* AND X(1).dimref = X(2).dimref;
 WHEN (3)

/* la relation logique contient 3 relations physiques */
* SELECT L
* FROM X(1),X(2),X(3)
* WHERE X(1).keyid = X(2).keyid = X(3).keyid
* AND X(1).dimref = X(2).dimref = X(3).dimref;
 WHEN (4)

/* la relation logique contient 4 relations physiques */
* SELECT L
* FROM X(1),X(2),X(3),X(4)
* WHERE X(1).keyid = X(2).keyid = X(3).keyid = X(4).keyid
* AND X(1).dimref = X(2).dimref = X(3).dimref = X(4).dimref;
 OTHERWISE
 PUT SKIP ("la relation logique contient plus de 4 relations physiques");
 END;

END PROJ;

```



#### 7.4.2.2 Sélection sur filtre

-----

Format général : SEL (R {L op V})

où R : nom de relation logique

L : liste de composantes de R

V : expression

op : opérateur de comparaison

{L op V} : filtre

La sélection sur filtre peut se réaliser grâce au programme suivant :

SEL : PROCEDURE;

déclarations

\* LET prphy BE

\*     SELECT relphid INTO rphy

\*     FROM RELPHY

\*     WHERE RELOGID = relation logique;

\* SELECT keyid,dimref

\*     FROM RELLOG

\*     WHERE RELOGID = relation logique;

\* OPEN prphy;

  I = 0;

  DO WHILE (code = ok);

\*     FETCH prphy;

      I = I + 1;

      X(I) = rphy;

  END;

\* CLOSE prphy;

/\* on ramène toutes composantes des relations physiques (de la relation logique) dans une relation logique en sélectionnant les tuples satisfaisant aux conditions exprimées dans le filtre \*/

SELECT I

  WHEN (1)

/\* la relation logique contient une relation physique \*/

\* SELECT \*

\*     FROM X(1)

\*     WHERE {L op V};

WHEN (2)

/\* la relation logique contient 2 relations physiques \*/

\* SELECT \*

\* FROM X(1),X(2)

\* WHERE X(1).keyid = X(2).keyid

\* AND X(1).dimref = X(2).dimref

\* AND {L op V};

WHEN (3)

/\* la relation logique contient 3 relations physiques \*/

\* SELECT \*

\* FROM X(1),X(2),X(3)

\* WHERE X(1).keyid = X(2).keyid = X(3).keyid

\* AND X(1).dimref = X(2).dimref = X(3).dimref

\* AND {L op V};

WHEN (4)

/\* la relation logique contient 4 relations physiques \*/

\* SELECT \*

\* FROM X(1),X(2),X(3),X(4)

\* WHERE X(1).keyid = X(2).keyid = X(3).keyid = X(4).keyid

\* AND X(1).dimref = X(2).dimref = X(3).dimref = X(4).dimref

\* AND {L op V};

OTHERWISE

PUT SKIP ('la relation logique contient plus de 4 relations physiques');

END;

END SEL;

#### 7.4.2.3 Jointure de deux relations

Format général : JOIN (R1 (L1) \* R2 (L2))

où R1,R2 : noms de relations logiques

L1,L2 : liste des composantes respectivement de R1,R2  
intervenant dans la jointure

La jointure de deux relations sur une liste de composantes peut se réaliser grâce au programme suivant :



JOIN PROCEDURE;

déclarations

```
* LET prphy BE
* SELECT relphid INTO rphy
* FROM RELPHY
* WHERE RELOGID = nom de relation R1;
* SELECT keyid,dimref
* FROM RELLOG
* WHERE RELOGID = nom de relation R1;
```

\* OPEN prphy;

I = 0;

DO WHILE (code = ok);

\* FETCH prphy;

I = I + 1;

X(I) = rphy;

END;

\* CLOSE prphy;

/\* 1ère étape: on fusionne les relations physiques de R1 en définissant une 'vue'

2ème étape: on fusionne les relations physiques de R2 en définissant une 'vue'

3ème étape: on fait la jointure entre les 2 images \*/

/\* 1ère étape \*/

/\*===== \*/

SELECT I

WHEN (1)

\* DEFINE VIEW R1(\*) AS

\* SELECT \*

\* FROM X(1);

WHEN (2)

/\* la relation logique R1 contient 2 relation physiques \*/

\* DEFINE VIEW R1(\*) AS

\* SELECT \*

\* FROM X(1),X(2)

\* WHERE X(1).keyid = X(2).keyid

\* AND X(1).dimref = X(2).dimref;

WHEN (3)

/\* la relation logique R1 contient 3 relations physiques \*/

\* DEFINE VIEW R1(\*) AS

\* SELECT \*

\* FROM X(1),X(2),X(3)

\* WHERE X(1).keyid = X(2).keyid = X(3).keyid

\* AND X(1).dimref = X(2).dimref = X(3).dimref;

WHEN (4)

/\* la relation logique R1 contient 4 relations physiques \*/

\* DEFINE VIEW R1(\*) AS

\* SELECT \*

\* FROM X(1),X(2),X(3),X(4)

\* WHERE X(1).keyid = X(2).keyid = X(3).keyid = X(4).keyid

\* AND X(1).dimref = X(2).dimref = X(3).dimref = X(4).dimref;

OTHERWISE

PUT SKIP ('la relation logique R1 a plus de 4 relations physiques');

END;

/\* 2ème étape \*/  
/\*===== \*/

ON REPRENDR L' ETAPE PRECEDENTE EN REMPLACANT R1 par R2

/\* 3ème étape \*/  
/\*===== \*/

/\* on ramène les 2 relations logiques en une relation en faisant une jointure  
sur une liste de composantes \*/

\* SELECT \*

\* FROM R1,R2

\* WHERE R1.L1 = R2.L2;

\* DROP VIEW R1,R2;

#### 7.4.2.4 Adjonction de nouvelles composantes

---

D' après notre documentation, Système R ne donne pas la possibilité à l' utilisateur de définir des fonctions horizontales au sein d' une relation logique, c' est à dire qu' il n' y a pas moyen d' ajouter une composante, qui est une fonction arithmétique d' autres composantes de cette relation.



#### 7.4.2.5 Autres opérations

---

- les opérateurs d' union, d' intersection et de différence sont disponibles dans le Système R (cfr. 7.2.2.1.6)
- il en est de même pour un certain nombre de fonctions verticales telles que la moyenne, le nombre d' éléments, ... sur lesquelles on peut définir des filtres (cfr. 7.2.2.1.5)
- la clause GROUP BY permet de réaliser des tris (cfr. 7.2.2.1.4)

On limitera ici l' énumération des opérations possibles sur la base de données car les possibilités ci-dessus semblent être les seules intéressantes dans le contexte de la base de données proposée par (BDER).

#### 7.4.2.6 Commandes composées

---

Jusqu' à présent, nous n' avons parlé que de commandes simples pour lesquelles on n' est pas obligé de mémoriser des résultats intermédiaires. L' utilisateur qui fait de la consultation selective pourra, toutefois, être contraint d' exprimer ses désirs à l' aide de commandes composées. Ces commandes composées sont des combinaisons des trois opérations de base : projection, sélection et jointure. Dans ce cas, le Système R doit transcrire la commande de façon interne après l' exécution de chaque commande élémentaire et transférer les données dans une relation provisoire qui sera gérée par le système. Dans ce contexte, il est probablement plus intéressant de définir une 'vue' pour chaque relation logique de telle manière que, lors des opérations de consultation, on ne manipule que des unités logiques : 'vues' (cfr. 7.2.2.3.3)

## 7.5 Application mise à jour

Il existe quatre opérations de base utilisées par le Système R pour la mise à jour de relations d' une base de données. Ce sont les suivantes :

- la modification de données existantes (UPDATE)
- l' ajoute de données (INSERT)
- la suppression de données (DELETE)
- la création d' une nouvelle relation regroupant des composantes d' autres relations (ASSIGN)

Nous nous intéressons, dans ce chapitre, plus particulièrement aux trois premières opérations.

### 7.5.1 Mise à jour ponctuelle de la base de données

Rappelons que mettre à jour la base données consiste, soit à mettre à jour le noyau, soit à mettre à jour l' entrepôt.

Le format général de la commande est le suivant :

MAJ nom de relation physique, mot de passe en écriture

Le type d' opération à faire dépendra de la valeur de la clé :

- si la clé n' existe pas il s' agit d' une insertion (INSERT)
- si la clé existe il s' agit d' une modification (UPDATE) sauf si l' utilisateur a positionné l' indicateur de suppression, il s' agit alors d' une suppression (DELETE)

Nous supposons la reconnaissance de l' utilisateur effectuée (SIGNON), le nom de l' utilisateur connu (userid) et la valeur de applic est égale à blanc pour la mise à jour (cfr. 6.6.1)

Dans la suite des programmes de mise à jour les lignes précédées par le caractère '\*' sont celles écrites en SEQUEL les autres sont écrites en PL/1.

'Programme' de mise à jour ponctuelle de la base de données

---

MAJ PROCEDURE;

déclarations

```
* SELECT relphid,userid,wrpass
* FROM RELPHY
* WHERE RELPHID = nom de relation physique
* AND WRPASS = mot de passe en écriture
* AND CALCULEE NOT = '1'
```



```
/* si le sélect a trouvé toutes les égalités on continue
 sinon on imprime un message d' erreur */
```

```
* SELECT relogid,mvt
* INTO relogid,mvt
* FROM RELLOG
* WHERE RELOGID = RELPHY.relogid
```

```
/* après avoir trouvé la relation logique correspondant à la relation physique
 examinons sa caractéristique de mouvement */
```

```
IF mvt = 'EX' THEN CALL MAJNOY ELSE
IF mvt = 'SIT' OR mvt = 'SIT' THEN CALL MAJENT
ELSE;
END MAJ;
```

#### 7.5.1.1 'Programme' de mise à jour ponctuelle du noyau

---

```
MAJNOY PROCEDURE;
```

```
déclarations
```

```
/* l' utilisateur entre le nom des composantes qu' il désire mettre à jour */
```

```
/* Système R vérifie automatiquement l' appartenance des composantes fournies
 à la relation à mettre à jour.
```

```
Système R vérifie également automatiquement le type et le format de chaque
composante. */
```

```
/* reconnaissance du type d' opération */
```

```
LECTURE : /* lecture d' un tuple fourni par le gestionnaire (clé et valeur des
composantes à mettre à jour.
```

```
Si c' est la fin de la mise à jour GO TO FIN */
```

```
/* lecture de l' indicateur de suppression */
```

```
* IF (clé existe) THEN IF INDIC-SUPP = 1 THEN DELETE nom de relation
* WHERE VALEUR DE LA CLE =
* valeur clé lue
* ELSE UPDATE nom de relation
* SET DATAID = valeur de la
* composante 1, ...?
* ELSE INSERT INTO nom de relation
* (DATAID 1,...,DATAID n) (valeur 1,...,valeur n);
```

```
GO TO LECTURE;
```

```
FIN : END MAJNOY;
```

### 7.5.1.2 'Programme' de mise à jour ponctuelle de l' entrepôt

---

MAJENT PROCEDURE;

déclarations

IF RELLOG.mvt = 'EV' THEN CALL MAJEV  
ELSE CALL MAJSIT;

END MAJENT;

#### 7.5.1.2.1 'Programme de mise à jour ponctuelle de données 'événement'

---

MAJEV PROCEDURE;

déclarations

/\* l' utilisateur entre le nom des composantes à mettre à jour \*/

/\* Système R vérifie automatiquement l' appartenance des composantes fournies  
à la relation à mettre à jour

Système R vérifie également automatiquement le type et le format de chaque  
composante \*/

/\* reconnaissance du type d' opération \*/

LECTURE : /\* lecture d' un tuple fourni par le gestionnaire (clé et valeur des  
composantes à mettre à jour

Si c' est la fin de la mise à jour GO TO FIN \*/

/\* lecture de l' indicateur de suppression \*/

IF (clé existe) THEN IF INDIC-SUPP = 1 THEN DELETE nom de relation  
WHERE VALEUR DE LA CLE =  
valeur clé lue  
ELSE UPDATE nom de relation  
SET DATAID = valeur de la  
composante 1, ...n

ELSE INSERT INTO nom de relation

(DATAID 1,...,DATAID n) (valeur 1,...,valeur n);

/\* le test de présence obligatoire est effectué  
automatiquement par Système R par la défini-  
tion préalable de TRIGGER cfr. ex 1 page  
suivante \*/

GO TO LECTURE;

FIN : END MAJEV;



EX 1 : exemple d' utilisation de la clause TRIGGER afin de réaliser automatiquement le TEST DE LA PRESENCE OBLIGATOIRE

```
DEFINE TRIGGER t1
 ON INSERT OF nom de relation
 CALL TPRESOBL;
```

La procédure TPRESOBL vérifiera si il y a lieu l' appartenance de la valeur des composantes non temporelles de la clé au noyau

EX 2 : exemple d' utilisation de la clause TRIGGER afin de réaliser automatiquement la vérification de l' appartenance d' une donnée à une formule de calcul et l' exécution du calcul si il y a lieu.

```
DEFINE TRIGGER t2
 ON UPDATE OR INSERT OF nom de relation
 CALL CALCUL;
```

La procédure CALCUL vérifiera l' appartenance d' une composante à une formule de calcul et appellera la routine de calcul si il y a lieu

#### 7.5.1.2.2 'Programme de mise à jour ponctuelle de données 'situation'

MAJSIT PROCEDURE:

déclarations

```
/* l' utilisateur entre le nom des composantes à mettre à jour */
/* Système R vérifie automatiquement l' appartenance des composantes fournies
 à la relation à mettre à jour
 Système R vérifie également automatiquement le type et le format de chaque
 composante */

/* reconnaissance du type d' opération */
LECTURE : cfr. 7.5.1.2.1 MISE à JOUR PONCTUELLE DE DONNEES 'événement'
/* Système R se charge automatiquement de vérifier si une donnée est à calculer
 et du calcul de celle-ci grâce à la définition préalable d' un
 TRIGGER cfr. ex 2 ci-dessus */

/* mise à jour éventuelle de l' état de la version de cette relation */
* SELECT pref,cardin
* INTO pref,cardin
* FROM VERSIONS
* WHERE RELPHID = nom de relation
* AND PREF = période de référence (déduite de la clé)
```

```

IF (select a trouvé égalités) THEN NBOCC = CARDIN + 1
ELSE;

* SELECT cardinal
* INTO card
* FROM KEYUNIT
* WHERE KEYID = RELLOG.KEYID

* IF NBOCC = card THEN UPDATE VERSIONS
* SET ETAT = 'c',CARDIN = NBOCC
* ELSE;
* IF NBOCC < card THEN UPDATE VERSIONS
* SET ETAT = 'p',CARDIN = NBOCC
* ELSE;

GO TO LECTURE;

END MAJSIT;

```

#### 7.5.2 Mise à jour ponctuelle de la base de données descriptive - extensibilité

L' étude de (SEQ2), (CHLD), (SYSR1), (SYSR2) et (ASLR) nous a révélé un manque de documentation pour réaliser l' implémentation de la mise à jour de la B.D.D. Nous examinons toutefois l' ajoute ou la suppression d' une relation (logique ou physique) et l' ajoute d' une composante à une relation.

##### 7.5.2.1 L' ajoute ou la suppression d' une relation (logique ou physique)

Malgré le manque d' informations au sujet de l' ajoute ou de la suppression d' une relation physique ou logique, nous proposons de créer un interface pour supprimer les informations, non seulement, dans la base de données mais également dans la B.D.D.

##### 7.5.2.2 L' ajoute d' une composante à une relation physique

Système R permet grâce à la clause EXPAND de réaliser l' ajoute d' une composante à une relation. Le format général est le suivant :

```

EXPAND TABLE nom de relation
 ADD COLUMN nom de la composante (type,format,...)

```

Notons que les composantes sont toujours ajoutées à la droite de la relation.



### 7.5.3 Mise à jour massive de l' entrepôt

Rappelons que la mise à jour massive de l' entrepôt consiste à ajouter un ensemble de tuples (version) à une relation physique.

Le format général de la commande est le suivant :

MAJM nom de relation,writepass,période de référence

Tout comme pour la mise à jour ponctuelle nous supposons la reconnaissance de l' utilisateur effectuée (SIGNON), le nom de l' utilisateur connu (userid) et la valeur de applic égale à blanc pour la mise à jour (cfr. 6.6.1); nous présentons, ci-dessous le 'programme' de mise à jour massive de l' entrepôt c' est à dire uniquement l' ajoute de données caractérisées 'événement' ou 'situation'.

MAJM PROCEDURE;

déclarations

```
* SELECT relphid;userid,wrpass
* FROM RELPHY
* WHERE RELPHID = nom de relation
* AND WRPASS = wrpass
* AND CALCULEE NOT = '1'
```

```
/* si le select a trouvé toutes les égalités on continue
 sinon on imprime un message d' erreur */
```

```
SELECT relogid,mvt
 INTO relogid,mvt
 FROM RELLOG
 WHERE RELOGID = RELPHY.RELOGID
```

```
/* après avoir trouvé la relation logique de la relation physique, examinons
 sa caractéristique de mouvement */
```

```
IF mvt = 'EV' THEN CALL MAJMEV
 ELSE
```

```
IF mvt = 'SIT' THEN CALL MAJMSIT
 ELSE PUT SKIP ('la relation relphid n" appartient pas à l'entre-
 pôt');
```

END MAJM;

### 7.5.3.1 'Programme' de mise à jour massive de données 'événement'

---

MAJMEV PROCEDURE;

déclarations

LECTURE : /\* lecture d' un tuple

si il n' y a plus de tuples à lire GO TO FIN \*/

/\* Système R se charge automatiquement d' effectuer le test de la  
présence obligatoire ainsi que de la vérification du type et du  
format des composantes lors de l' ajoute \*/

\* UPDATE nom de relation

\* SET DATAID 1 = valeur 1, DATAID 2 = valeur 2, ..... DATAID n = valeur n

GO TO LECTURE;

FIN:END MAJMEV;

### 7.5.3.2 'Programme' de mise à jour massive de données 'situation'

---

MAJMSIT PROCEDURE;

déclarations

LECTURE : /\* lecture d' un tuple

si plus de tuple à lire GO TO SUITE \*/

/\* Système R se charge automatiquement de :

- effectuer le test de la présence obligatoire
- vérifier le type et le format de chaque composante  
à mettre à jour
- contrôler si il y a des données à calculer \*/

\* UPDATE nom de relation

\* SET DATAID 1 = valeur 1, DATAID 2 = valeur 2, ....., DATAID n = valeur n

SUITE : /\* mise à jour éventuelle de l' état de la version de cette relation \*/  
(cfr. MAJSIT 7.5.1.2.2)

END MAJMSIT;



## 7.6 Conclusions de l' étude du Système R

L' étude du Système R s' est limitée à la lecture de différents articles (cfr. (SEQ2), (CHLD), (SYSR1), (SYSR2), (ASLR)), dans lesquels était présenté soit le Système R, soit son langage associé : SEQUEL. Nous n' avons, en effet, ni travaillé avec le Système R, ni eu à notre disposition un manuel utilisateur. Toutefois, si cette documentation ne nous a pas permis de tirer des conclusions au sujet des performances du Système R, nous insisterons sur les trois points suivants :

- facilités de description des données
- clause VIEW et clause TRIGGER
- système général de base de données.

### 7.6.1 Facilités de description des données

Le prototype Système R offre beaucoup de facilités pour la description des données. Ainsi, pour la consultation, ses tables de description reprennent la quasi-totalité des données de la base de données descriptive spécifiques à cette application de consultation. Pour l' application de mise à jour on doit écrire des programmes relativement simples.

Pour réaliser les deux applications définies sur la base de données, celle de consultation et de mise à jour, le Système R devrait utiliser, outre ses tables de description, quelques relations de la base de données descriptive afin d' assurer la cohérence de la base de données :

```
RELLOG (RELOGID,DIMREF,MVT,NBMXVR,KEYID)
RELPHY (RELPHID,USERID,WRPASS,RELOGID,CALCULEE)
KEYUNIT (KEYID,CARDINAL)
DATATYPE (DATAID,PRIMAIRE)
COMDATA (DATAID,FORMULE,PROGR)
VERSIONS (RELPHID,PREF,ET T,CARDIN)
COMPOSANT (REPHID,DATAID,NUMDATA)
```

### 7.6.2 Clause VIEW et clause TRIGGER

Parmi les idées que propose le Système R, deux nous semblent très intéressantes dans le contexte de la base de données (BDER) : la clause VIEW et la clause TRIGGER. La VIEW permet de regrouper, dans notre cas, les relations physiques d' une relation logique; étant donné que nous n' avons aucune infor-

mation quant à sa réalisation technique (redondance ou non des données par exemple) il nous est difficile de juger de son efficacité.

Le TRIGGER permet de déclencher un certain nombre d' actions si on effectue une opération de mise à jour sur des données (UPDATE, DELETE, INSERT). Ainsi, par exemple, pour la mise à jour de données primaires on définit un TRIGGER pour déclencher l' appel à la routine de calcul.

### 7.6.3 Système général de base de données

Comme tout système général de base de données, qui ne veut pas se limiter à un problème bien particulier, le prototype Système R offre beaucoup de facilités qui ne sont pas indispensables pour la base de données économique régionale. De plus, étant donné la philosophie générale de (BDER) qui propose la constitution d' un ensemble d' informations avec un maximum de données mais un minimum d' opérations, il nous semble plus intéressant de définir un système général de base de données bien adapté à ce problème et par conséquent, le plus performant possible. Celui-ci ne pourra être utilisé pour gérer d' autres bases de données qui n' ont pas la même particularité.



## CONCLUSION GENERALE

Le but de notre mémoire était de réaliser l'implémentation physique d'une base de données économiques régionale en nous basant sur des travaux antérieurs concrétisés par (BDER). Cette réalisation a été retardée, non seulement par un manque de rigueur dans l'application des critères de structuration des données, mais également par la nécessité de redéfinir et d'ajouter des critères de découpe aux différents niveaux de structuration.

C'est ainsi, qu'au niveau infologique, nous nous sommes inspirés du modèle 'entité-association' pour définir des types d'entités et d'associations auxquels on associe un ou plusieurs types de propriétés : identifiant unique, c'est à dire que deux entités du même type ont des valeurs distinctes pour cet identifiant. Ce modèle sera, donc, formé d'un ensemble de types d'entités et d'associations ayant un ensemble de types de propriétés (dont cet identifiant unique), et possédant, par ailleurs, certaines caractéristiques qui serviront ultérieurement en tant que critères de structuration.

Lors du passage du niveau infologique au niveau conceptuel, c'est à dire dans notre cas, le passage du modèle 'entité-association' au modèle relationnel, un type d'entité ou d'association est découpé en relations

ayant à la fois un même identifiant unique, la même caractéristique de mouvement et une dimension de référence identique. A ce stade, (BDER) n'a pas tenu compte du critère de mouvement, ce qui se justifie si on s'est mis d'accord sur des valeurs particulières de la dimension de référence pour des données qui ont comme caractéristique de mouvement 'événement' ou 'existant'. En toute généralité, il vaut cependant mieux accepter notre proposition car la dimension de référence n'est utilisée explicitement que pour les données de caractéristique de mouvement 'situation'.

La structuration, au niveau logique, s'effectue en fonction des deux applications définies sur la base de données : la mise à jour et la consultation. A chaque relation définie au niveau conceptuel on fait correspondre une et une seule relation logique qui sera l'unité de base de la consultation. Les données d'une relation logique recouvrent le même champ sémantique et ont une forte probabilité d'être demandées en même temps, car cette relation a un sens pour l'utilisateur.



Pour l' application de mise à jour, les relations logiques sont scindées en relations physiques en fonction de trois critères : il est, en effet, nécessaire, sous peine de s' obliger à de multiples manipulations lors de la mise à jour, d' utiliser les critères de source et de fréquence de collecte des données. De plus, le critère de 'donnée calculée' permet d' interdire la mise à jour de relations physiques ne contenant que des données calculées.

A ce niveau, on découpe la base de données en deux parties : l' entrepôt et le noyau. L' entrepôt est un ensemble de relations contenant des données auxquelles on accède massivement et séquentiellement, tandis que les relations du noyau, en général de type signalétiques, sont plutôt utilisées sélectivement et seront donc implémentées en accès direct.

Au niveau de l' implémentation physique nous avons effectué quelques choix tels que l' incorporation de la notion de version aux relations et la correspondance biunivoque entre un fichier et une relation physique. Nous avons, ensuite, décrit les algorithmes de consultation et de mise à jour qui gèrent la base de données à l' aide d' une base de données descriptive relationnelle. Celle-ci est définie, comme un ensemble de douze relations, dans (BDER). Cependant, nous y avons apporté des modifications dues, d' une part, aux nouveaux critères de structuration et d' autre part, à des incohérences qui nous ont entraîné, par exemple, à transférer la composante READPASS (mot de passe en lecture) de la relation RELPHY à la relation RELLOG de la B.D.D..

Ayant fait cette mise au point, nous présentons les deux formes de consultation : tout d' abord, la plus simple d' entre elles, la consultation massive, permet d' obtenir une relation logique telle qu' elle existe dans la base de données c' est à dire avec toutes les occurrences de ses composantes; ensuite la consultation sélective offre la possibilité à l' utilisateur d' éliminer des données dont il n' a pas besoin. Pour ce faire, nous distinguons trois opérations de base :

- la projection d' une relation sur une liste de composantes
- la jointure de deux relations sur une liste de composantes
- la sélection d' occurrences au moyen d' un filtre. L' application de consultation peut être effectuée sur la base de données et sur la base de données descriptive.



L' application de mise à jour comporte deux types d' opérations : d' une part la mise à jour massive de l' entrepôt qui consiste, en général, en l' apport d' une ou plusieurs versions de données à une relation physique, et d' autre part la mise à jour ponctuelle de la base de données et de la base de données descriptive, qui ne manipule qu' un nombre limité d' occurrences de relations; cette deuxième opération est réservée au gestionnaire de la base de données afin d' assurer la cohérence de celle-ci. Les traitements pour la mise à jour de la base de données diffèrent selon la caractéristique de mouvement ainsi, par exemple, il est nécessaire d' effectuer le test de la présence obligatoire lors de la mise à jour de données de caractéristique de mouvement 'événement' ou 'situation'.

La mise à jour de la base de données descriptive nous a surtout intéressé parce qu' elle rend possible l' extensibilité du modèle : il doit, en effet, être possible d' ajouter ou de supprimer des relations logiques et physiques ainsi que des composantes avec un maximum de souplesse.

Ayant défini tous ces algorithmes, nous avons programmé, lors de notre stage, ceux qui concernent la mise à jour de la base de données. Pour gérer les données, nous avons utilisé toutes les composantes de la B.D.D. spécifiques à cette application tout en éliminant celles qui étaient reprises dans le descriptif existant. Celui-ci a été défini afin de pouvoir consulter la base de données par l' intermédiaire d' un langage d' interrogation de bases de données : LAPIN. Nous avons utilisé celui-ci car il nous était impossible, dans le cadre de notre mémoire, de définir notre propre langage. Nous avons préféré nous consacrer entièrement à la mise à jour, car c' est l' opération qui pose le plus de problèmes dans les modèles relationnels.

L' opération de mise à jour massive est, bien entendu, l' opération la plus efficace; cependant la mise à jour ponctuelle est indispensable afin de pouvoir, par exemple, corriger des valeurs ou modifier la structure de la base de données.

L' opération de consultation nous a permis d' étudier un prototype de langage qui n' était, à ce moment là, pas particulièrement adapté à nos besoins étant donné qu' il n' était pas possible de réaliser certaines opérations (jointure sur une liste de composantes non adjacentes) alors que d' autres (fonctions verticales), qui ne jouent qu' un rôle secondaire dans l' application consultation, sont disponibles.

Après avoir utilisé réellement un système existant, nous avons fait l'étude théorique d'un autre système général de bases de données : le Système R. Il est plus complet que LAPIN car il permet la mise à jour de données. De plus, toutes les données de la B.D.D. concernant l'application de consultation peuvent être supprimées car elles sont gérées automatiquement par Système R à condition de définir une 'vue' pour chaque relation logique. Pourtant, le Système R est trop sophistiqué pour être bien adapté à chaque problème particulier. C'est pourquoi, il semble préférable d'employer un système de gestion propre à la base de données économiques régionale. Ce choix se justifie surtout par la philosophie générale de ce type de base de données qui est de stocker le maximum d'informations avec un minimum de manipulation.

Nous sommes conscients du fait que l'implémentation de cette base de données virtuelle n'est pas réalisable pour l'instant étant donné le contexte socio-politique, cependant, nous espérons avoir suffisamment fait mûrir le projet qui nous a été confié afin qu'il soit possible de le concrétiser si les mentalités changeaient.



## REFERENCES

=====

- (ASLR) : R.A. LORIE and J.F. NILSON *An Access Specification Language for a Relational Database System* IBM Research Laboratory San Jose, CA
- (BBBC) : BENCI, E., BODART, F., BOGAERT, H., CABANES, A., *Concepts for the Design of a Conceptual Schema* in 'Modeling in Data Base Management Systems' G.M. Nijssens ed. Proceedings of the IFIP Working Conference, Freudenstadt, Germany, 5-8 January 1976, North Holland Publishing Company 1976
- (BDER) : J. BERLEUR *Une Banque de Données Economiques Régionale ? Aspects économique, informatique, juridique et socio-politique* Travaux de l' Institut d' Informatique n° 5 dans le cadre du Séminaire 'L' Homme et l' Informatique' sous la Direction de J. BERLEUR
- (CBDH) : J.L. HAINAUT *Base de Données* cours de 2ème Licence Institut d' Informatique Namur
- (CBTL) : F. BODART *Problèmes d' organisation et méthodes d' Analyse Fonctionnelle* cours de 3ème Licence Institut d' Informatique Namur
- (CHLD) : R.A. LORIE and B.W. WADE *The Compilation of a Very High Level Data Language* IBM Research Laboratory San Jose California
- (CODD1) : E.F. CODD *A Relational Model of Data for Large Shared Data Banks* Comm ACM 13,4 (june 1970) pp. 377-387
- (CODD2) : E.F. CODD *Normalized Data Base Structure a Brief Tutorial* (p. 1-17) ACM-SIGFIDET 1971
- (DESCM) : Brochure *Content of DESCMS* IBM Centre Scientifique Paris
- (DESCR) : J.P. ADAM *Description technique du système de mise à jour* IBM Centre Scientifique Paris 1978
- (GIRL) : Brochure *GIRL : General Interface with Relational Logic* IBM Centre Scientifique Paris
- (LAPIN) : O. BERTRAND, J.J DAUDENARDE, B. du CASTEL *USAGE : a User Generator* Second Draft pp. 14-31 IBM France Scientific Center Paris
- (OCBD) : E. COLLARD, C. PETIT *Organisation Conceptuelle d' une Base de Données Economiques à Usage d' Organismes Définisseurs de Politique Economique Régionale* Mémoire de Licence et Maîtrise en Informatique Institut d' Informatique Namur 1976

REFERENCES (suite)

=====

- (SEQ2) : D.D. CHAMBERLIN et al. *SEQUEL 2 : A Unified Approach to Data Definition, Manipulation and Control* IBM Journal of Research and Development 20,6 November 1976 pp. 560-575
- (SYSR1) : M.M. ASTRAHAN et al. *SYSTEM R : A Relational Approach to Database Management* ACM Transactions on Database Systems Vol 1 n° 2 (june 1976) pp. 97-137
- (SYSR2) : M.M. ASTRAHAN et al. *SYSTEM R : Relational Database Management System* Proc IEE Compcon 79 Spring, February 26-March 1, San Francisco



## BIBLIOGRAPHIE SOMMAIRE

- ASTRAHAN, M.M., ALTMAN E.B., FELDER P.L., SENKO M.E. *Concepts of Data Independent Accessing Model* Proc. ACM-SIGFIDET Work. 1972
- ASTRAHAN, M.M. and CHAMBERLIN D.D. *Implementation of a Structured English query language* Comm. ACM 18,10 (oct 1975) pp. 580-587
- BRACHMAN, C.W. *The data structure set model* in Proc. 1974 ACM SIGMOD Debate 'Data Models : Data Structure-Set Versus Relational' R. Rustin (Ed.), ACM, New York, 1975, pp. 1-10
- BAYER, R. and McCREIGHT, E. *Organization and maintenance of large ordered indices* Proc. ACM-SIGFIDET Workshop Nov. 1970, ACM, New York, 1970 pp. 107-141
- BELL, C.J. *A Relational Model for Information Retrieval and the Processing of Linguistic Data* IBM Research Report RC 1975
- BLASGEN, M. and ESWARAN, K. *On the Evaluation of Queries in a Relational Data base System* IBM Research Report RJ1745 San Jose April 1976
- BOYCE, R.F. and CHAMBERLIN, D.D. *Using a Structured English query Language as a data definition facility* Report RJ1318 IBM T. Watson Research Center Yorktown Heights, New York, Dec. 1973
- BRACCHI, G., FEDELI, A. and PAOLINI, P. *A language for a Relational Data base* Sixth Annual Princeton Conf. on Information Sciences and Systems, March 1972 Princeton Univ., N.J. 1972
- CHAMBERLIN, D.D. *Relational data-base management systems* in Computing Surveys, Vol. 8 n° 1, March 1976, pp. 43-66
- CHIBA, Y. *A data base search algorithm based on complicated retrieval conditions* The Soken Kiyo 5,1 (1975) pp. 159-175
- CODASYL Data Base Task Group, April 1971 Report, ACM, New York, 1971
- CODD, E.F. *A data base sublanguage founded on the relational calculus* in Proc 1971 ACM-SIGFIDET Work. on Data Description, Access and Control, ACM, New York pp. 35-65
- *Further normalization of the data base relational model* Courant computer Science Symposia 6, Data Bases Systems, New York, May 1971, Prentice-Hall, New York, 1971, pp. 33-64
- CODD, E.F. and DATE, C.J. *Interactive support for non-programmers : the relational and network approaches* Proc. 1974 ACM-SIGMOD op. cit.

- COOK, T.J. *A data base management system design philosophy* in Proc. 1975 ACM-SIGMOD International Conf. on Management of Data, ACM, New York pp. 15-22
- DATE, C.J. and HOPEWELL, P. *Storage structure and physical data independence*, Proc 1971 ACM-SIGFIDET op. cit. ACM, New York 1971, pp. 139-168
- DEE, E., HILDER, W., KING, P.J. and Taylor, E. *Cobol extensions to handle a relational data base* British Computer Society, Working Party 5, Oct. 1973
- DELOBEL, C. *Aspects théoriques sur la structure de l'information dans une base de données* Revue Française d'Informatique et de Recherche Opérationnelle, Sept. 1971
- EARNEST, C.P. *A comparaison of the network and relational data structure models* Technical Report, Computer Sciences Corp. El Segundo, Calif. April 1974
- FARLEY, J.H., GILLES and SCHUSTER, S.A. *Query execution and index selection for relational data bases* Technical Report CSRG-53 Computer Systems Research Group, Univ. of Toronto, Toronto Canada, March 1975
- GOTELEB, L.R. *Computing joins of relations* in Proc. 1975 ACM-SIGMOD op. cit. ACM, New York pp. 55-63
- HALL, P.A. *Optimisation of a single relational expression in a relational data base system* IBM Scientific Centre Report UKSC 0076, Peterlee, England July 1975
- HANSAL, A. *A formal definition of a relational database system* IBM UK Scientific Centre Report, Peterlee, England, June 1976
- HUITS, M. *Requirements for language in data-base systems* in Data Base Description DOUQUE, B.C. and Nijssens (Ed.) North-Holland Publ. Co., Amsterdam, The Netherlands, 1975 pp. 85-100
- HAINAUT, J.L., LECHARLIER, B. *An extensible Semantic Model of Data Base and its Data Languages* Information Processing 74, North Holland Publishing Company (1974)
- LORIE, R.A., and SYMONDS, A.J. *A relational access method for interactive applications* in Courant Computer Science Symposia, Vol 6 op. cit. 1971 pp. 99-124
- MELTZER, H.S. *Relations and relational operations* IBM Report to GUIDE 38 Information Systems Division, Dallas Texas, May 1974



- NIJSSSEN, G.M. *Data structuring in the DDL and relational model* in 'Data Base management' J.W. Klimbie and K.L. Koffeman (Ed.) North-Holland Publ. Co. Amsterdam, The Netherlands, 1974 pp. 363-384
- PIROTTE, A. and WODON, P. *A comprehensive formal query language for a relational data base : FQL* Technical Report R283 M.B.L.E. Laboratoire de Recherches, Brussel, Belgium, Dec. 1974
- ROTHNIE, J.B. *Evaluating interentry retrieval expressions in a relational data base management system* in Proc. AFIPS 1975 National Computer Conf. AFIPS Press, Montvale, N.J. pp. 417-423
- STACEY, G.M. *The interface between a database and its host languages* in Data Base management op. cit. The Netherlands, 1974 pp. 305-312
- STONEBRAKER *INGRES - a relational data base system* in AFIPS Conf. Proceedings Vol. 14 pp. 409-415
- *Implementation of integrity constraints and views by query modification* in Memorandum n° ERL-M514, 1975, March 17
  - *The design and implementation of INGRES* in Memorandum n° ERL-577, Jan. 1976
  - *Storage structures and access methods in the relational data base management system INGRES* in Memorandum n° ERL-505, March 1975
  - *The INGRES Protection System* in Memorandum n° ERL-M594, July 1976
  - *Creating and Maintaining a database using INGRES* in Memorandum n° UCB/ERL M77/71, Dec. 1977
- TODD, S.J. *Peterlee relational test vehicle PRTV, a technical overview* IBM Scientific Centre Report UKSC 0075, Peterlee, England, July 1975
- WITHNEY, V.K. *A relational data management implementation techniques* in Proc. 1974 ACM-SIGMOD op. cit. pp. 321-348
- WINOGRAD, T. *Procedures as a representation for data in a computer program for understanding natural language* MIT Project MAC Report MAC TR-84, 1971 Cambridg., Mass
- WINSLOW, L.E. *An efficient implementation of CODD's relational model data base* Proc. COMPCON 75, 11th Annual IEEE Computer Society Conf. Sept. 1975
- ZLOOF, M.M. *Query by Example* Research Report RC4917 IBM T.J. Watson Research Centre, Yorktown Heights, N.Y., July 1974

ENSEMBLE DES RELATIONS DE LA  
=====

BASE DE DONNEES  
=====



On trouvera ci-après l' ensemble des relations logiques et physiques qui constituent la base de données.

Légende :

- Un trait continu \_\_\_\_\_ délimite les "relations logiques"
- Un trait discontinu ----- délimite les "relations physiques"

Signification des colonnes :

- [1] : numéro de la relation logique
- [2] : champ couvert par la relation logique
- [3] : nom de la relation logique
- [4] : caractéristique de mouvement
- [5] : dimension de référence
- [6] : composantes primaires non temporelles
- [7] : nom des composantes primaires non temporelles
- [8] : numéro de la relation physique
- [9] : nom de la relation physique
- [10] : source
- [11] : fréquence de collecte
- [12] : critère de calcul
- [13] : nom des composantes non primaires
- [14] : numéro des composantes
- [15] : intitulé des composantes

Liste des abréviations utilisées pour [4] : caractéristique de mouvement

- EX : existant
- EV : événement
- SIT : situation

Liste des abréviations utilisées pour [5] : dimension de référence

- AN : année
- TRIM : trimestre
- MOIS : mois

Liste des abréviations utilisées pour [10] : source

---

|                |                                                                           |
|----------------|---------------------------------------------------------------------------|
| BNB            | : Banque Nationale de Belgique                                            |
| Bourse         | : Bourse                                                                  |
| CGER           | : Caisse Générale d' Epargne et de Retraite                               |
| CMCES          | : Comité Ministériel pour la Coordination Economique et Sociale           |
| Créd. Comm.    | : Crédit Communal de Belgique                                             |
| INAMI          | : Institut National de l' Assurance Maladie Invalidité                    |
| INS            | : Institut National de Statistique                                        |
| INS-SPA        | : Formulaires de l' INS, relatifs aux statistiques de production annuelle |
| MAE            | : Ministère des Affaires Economiques                                      |
| MB             | : Moniteur Belge                                                          |
| Min. Cl.Moy.   | : Ministère des Classes Moyennes                                          |
| Min. Fin.      | : Ministère des Finances                                                  |
| Min. Sant.Pub. | : Ministère de la Santé Publique                                          |
| ONEM           | : Office National de l' Emploi                                            |
| ONSS           | : Office National de la Sécurité Sociale                                  |
| Org. Presse    | : Organismes de Presse                                                    |
| RAPPANNEP      | : Rapport Annuel d' Entreprise                                            |
| R.C.           | : Registre du Commerce                                                    |
| RN             | : Registre National                                                       |
| SEER           | : Secrétariat d' Etat à l' Economie Régionale                             |
| SNCI           | : Société Nationale de Crédit à l' Industrie                              |
| SNI            | : Société Nationale d' Investissement                                     |

Liste des abréviations utilisées pour [11] : fréquence de collecte

---

|      |                 |
|------|-----------------|
| SANS | : sans          |
| INEX | : inexistante   |
| VAR  | : variable      |
| AN   | : annuelle      |
| TRIM | : trimestrielle |
| MOIS | : mensuelle     |



Signification de [12] : critère de calcul

---

- 1 : indique que la relation physique ne contient que des composantes calculées. (excepté pour les composantes formant la clé)
- 0 : indique que la relation physique ne contient pas de composantes calculées.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]

1 Identifiant de l'établissement  
 IDET EX 1d. ET ET 1.1 IDET1 R.C. SANS 0 NOM 110001 Nom de ET  
 110002 Adresse de ET  
 DATECR 110003 Date création ET, date mise en activité  
 EP 110004 Identifiant EP dont dépend ET, n° TVA  
 SECTACTn 12010N Secteur d'activité N de ET (S'N)

1.2 IDET2 Sans INEX 0 DATEDEX 210001 Date de début d'exercice

2 Identifiant de l'entreprise  
 IDEP EX 1d. EP EP 2 IDEP R.C. SANS 0 ETHR 110005 Identifiant des ET hors région  
 110006 Adresse EP (siège social)  
 NOM 110007 Nom ou raison sociale EP  
 DATECR 110008 Date création EP (publication au Moniteur  
 FJUR 110009 Forme juridique EP  
 SECTACTn 12000N Secteur d'activité N de EP (S'N)

3 Dettes de l'entreprise  
 DETTES SIT TRIM 1d. EP EP 3.1 DONSS ONSS TRIM 0 MOISRO 440001 Nombre de mois de retard de paiements vis-à-vis de  
 1' ONSS  
 MONTANTO 440002 Montant des paiements en retard vis-à-vis de  
 1' ONSS

3.2 DPP Min. Fin. TRIM 0 MOISRP 440003 Nombre de mois de retard de paiements au Min. Fin.  
 pour précompte professionnel  
 MONTANTP 440004 Montant des paiements en retard vis-à-vis du  
 Min. Fin. pour préc. profes.

3.3 DIS Min. Fin. AN 0 MOISRI 440005 Nombre de mois de retard de paiements au Min. Fin.  
 pour impôt des sociétés  
 MONTANTI 440006 Montant des paiements en retard vis-à-vis du  
 Min. Fin. pour impôt des sociétés

4 Licenclement  
 LICENC SIT AN 1d. EP EP 4 LICENC ONEM VAR 0 NBLIC 440007 Nombre de personnes licenciées collectivement au  
 cours de 1' année

5 Capital passif  
 CAPASSIF SIT AN 1d. EP EP 5 CAPASSIF RAPPANNEP AN 0 VALPAS 422017 Valeur totale des postes de passif



|       |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
|-------|-----------------------|--------|--------|-------------------------------------------------|--------------|-----|-------|--------------|--------|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6     | Capital passif propre | CAPP   | SIT AN | Id. EP                                          | EP           | 6.1 | CAPP1 | RAPPANNEP AN | 0      | FPERM<br>FPRO<br>CAPSOC<br>CAPLIB<br>RES<br>PROV<br>REPORTS<br>BENET | 422001 Valeur globale des fonds permanents<br>422002 Valeur globale des fonds propres<br>422003 Valeur du capital social<br>422004 Valeur du capital libéré<br>422005 Valeur des réserves<br>422006 Valeur des provisions<br>422007 Valeur du report à nouveau<br>422008 Valeur du bénéfice net de l' exercice                                                                               |
| ----- |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
| 6.2   | CAPP2                 | SANS   | INEX 0 | SUBEQ<br>ACTN<br>ACTA<br>APPACT<br>RESPR        |              |     |       |              |        |                                                                      | 422009 Valeur des subventions d' équipement reçues<br>422020 Nombre d' actionnaires pour les actions nominatives<br>422024 Nombre de personnes ayant déposé des actions à l' Assemblée Générale Extraordinaire<br>422028 Part des apports d' actionnaires dans la dernière augmentation de capital<br>422029 Part du financement par réserve propre dans la dernière augmentation de capital |
| ----- |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
| 6.3   | CAPP3                 | Bourse | AN     | 0                                               | MOYACT       |     |       |              |        |                                                                      | 424009 Valeur moyenne du cours de l' action sur un trimestre                                                                                                                                                                                                                                                                                                                                 |
| ----- |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
| 6.4   | CAPP4                 | Bourse | AN     | 1                                               | EV8          |     |       |              |        |                                                                      | 424010 Indice d' évolution du cours de l' action en bourse                                                                                                                                                                                                                                                                                                                                   |
| ----- |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
| 7     | Capital passif propre | CAPP5  | SIT AN | Id. EP,<br>Id. TYPE<br>d' ACT.                  | EP<br>TYACT  | 7   | CAPP5 | SANS         | INEX 0 | NBACT<br>VALNOM                                                      | 422018 Nombre d' actions<br>422019 Valeur nominale de l' action                                                                                                                                                                                                                                                                                                                              |
| ----- |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
| 8     | Capital passif propre | CAPP6  | SIT AN | Id. EP,<br>Id. Action                           | EP<br>ACTNOM | 8   | CAPP6 | SANS         | INEX 0 | NAT<br>STATUT<br>NBACT                                               | 422021 Nationalité<br>422022 Statut<br>422023 Nombre d' actions souscrites                                                                                                                                                                                                                                                                                                                   |
| ----- |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
| 9     | Capital passif propre | CAPP7  | SIT AN | Id. EP, Id<br>EP ayant<br>parts 5%<br>dans Cap. | EP<br>EPPART | 9   | CAPP7 | SANS         | INEX 0 | PCPART                                                               | 422031 Valeur en % de la participation dans le capital de l' EP                                                                                                                                                                                                                                                                                                                              |
| ----- |                       |        |        |                                                 |              |     |       |              |        |                                                                      |                                                                                                                                                                                                                                                                                                                                                                                              |
| 10    | Capital passif propre | CAPP8  | SIT AN | Id. EP 00<br>SNI a des<br>parts                 |              | 10  | CAPP8 | SNI          | VAR 0  | VAL                                                                  | 414029 Valeur des participations de la SNI dans le capital de l' EP                                                                                                                                                                                                                                                                                                                          |

11 Capital passif CAPE SIT AN 1d. EP EP 11.1 CAPE1 RAPPANEP AN 0 VMLT 422010 Valeur globale des dettes à moyen et long terme  
VCT 422016 Valeur globale de l' exigible à court terme

11.2 CAPE2 SANS INEX 0 VEO 422011 Valeur des emprunts obligataires  
VENOP 422012 Valeur des emprunts non obligataires privés  
VEPC 422013 Valeur des emprunts auprès des ET publics de crédit  
VELTMB 422014 Valeur des emprunts à long terme sur le marché belge  
VELTME 422015 Valeur des emprunts à long terme sur le marché étranger

12 Capital passif CAPE3 EV 1d. EP, EP 12 CAPE3 Min. Fin. VAR 0 MONTANT 422032 Montant  
étranger 1d. EMP. EMP0BL TINT 422033 Taux d' intérêt  
OBLIG. TERME 422034 Terme  
DEMISS 422035 Date d' émission  
NBOBL 422036 Nombre d' obligations

13 Capital passif CAPE4 EV 1d. EP, EP 13 CAPE4 SNCI,CNCP VAR 0 MONTANT 422037 Montant  
étranger 1d. Autres AEMP TINT 422038 Taux d' intérêt  
EMP. EMP. Cred.Comm TERME 422039 Terme  
DEMISS 422040 Date d' émission  
ORGAN 422041 Organisme créancier

14 Capital actif CAPACTIF SIT AN 1d. EP EP 14 CAPACTIF RAPPANEP AN 0 VALACT 421010 Valeur totale des postes d' actif

15 Capital actif CAPANIMM SIT AN 1d. EP EP 15.1 CAPANIMM1 RAPPANEP AN 0 FCIRC 421007 Valeur globale des fonds circulants  
non-immobilisé FDISP 421008 Valeur des fonds disponibles  
FREAL 421009 Valeur des fonds réalisables  
TITRES 421011 Valeur globale du portefeuille-titres

15.2 CAPANIMM2 SANS INEX 0 BONETAT 421012 Valeur en bons d' Etat dans le portefeuille-titres  
PLPROV 421013 Valeur en placements provisionnaires (actions, oblig.), dans le portefeuille-titres

16 Capital actif CAPANIMM1 SIT AN 1d. EP EP 16 CAPANIMM1 RAPPANEP AN 0 IMMOB 421001 Valeur globale des immobilisations  
immobilisé ICORP 421002 Valeur des immobilisations corporelles  
figurant au PARTIC 421003 Valeur des participations  
bilan AUIMM 421004 Valeur des autres immobilisations  
ANIMM 421005 Amortissement des valeurs immobilisées  
AMIC 421006 Amortissement des immobilisations corporelles



17 Capital actif immobilisé fig. au bilan      CAPAIMM2 SIT AN      1d. EP, 1d EP ou elle EPPART a des parts      17      CAPAIMM2 SANS      INEX 0 PCPART      421015 Valeur en % de la participation dans le capital d' autres EP

18 Stock de capital physique      CAPHYS1 SIT AN      1d. ET      ET      18      CAPHYS1 SANS      INEX 0 VALBAT NBBAT SURF AGE VMATR VEQUOT VSMOY      321001 Valeur du stock de capital bâtiment 321002 Nombre de bâtiments 321003 Surface d' occupation des bâtiments 321004 Age des bâtiments 321005 Valeur du stock capital matériel roulant 321008 Valeur du stock capital équipement & outillage 321011 Valeur stocks moyens (prod. entr. en-cours, sort.)

19 Stock capital physique      CAPHYS2 SIT AN      1d. ET, 1d. Mat.      ET      MATR      19      CAPHYS2 SANS      INEX 0 NBMR      321006 Nombre d' unités

20 Stock de capital physique      CAPHYS3 SIT AN      1d. ET, 1d U. Mat.      ET      UMR      20      CAPHYS3 SANS      INEX 0 AGE      321007 Age

21 Stock de capital physique      CAPHYS4 SIT AN      1d. ET, 1d Ty. Eq.      ET      TEQ      21      CAPHYS4 SANS      INEX 0 NBUN      321009 Nombre d' unités

22 Stock de capital physique      CAPHYS5 SIT AN      1d. ET, 1d Un. Eq.      ET      UEQ      22      CAPHYS5 SANS      INEX 0 AGE      321010 Age

23 Ratios financiers      RATFIN SIT AN      1d. EP      EP      23.1 RATFIN1 RAPPANNEP AN      1      RENTCP SOLVMTL INDFIN CAPREMB RAMIMM FROUL ROTCAP MARGEGR      425001 Coefficient de rentabilité du capital propre 425003 Coefficient de solvabilité à moyen. long terme 425004 Coefficient d' indépendance financière 425006 Indice de capacité de remboursement 425009 Rapport entre amortissements et immobilisations 425010 Coefficient de fonds de roulement 425011 Coefficient de rotation du capital 425012 Marge bénéficiaire brute

23.2 RATFIN2      INS, RAPPANNEP      AN      1      VATMM TAFIN      425002 Valeur ajoutée produite par immobilisations 425008 Taux d' autofinancement des investissements

23.3 RATFIN3      ONSS, INS Min. Fin.      AN      1      RCTA      425016 Rapport du coût du travail/personne au chiffre d' affaires/personne

23.4 RATFIN4 SANS INEX 1 IEND 425005 Indicateur du poids de 1' endettement  
 ICAPREN 425007 Indice de capacité bénéficiaire  
 VANH 425013 Valeur des annuités  
 PANNCFN 425014 Part des annuités dans le cash flow net  
 RAMAN 425015 Rapport entre les amortissements et les annuités

24 Capital investi CAPINV SIT AN id. ET ET 24.1 CAPINV1 INS AN 0 VINV 324001 Valeur investissement global effectué durant l'année  
 VPREV 324009 Valeur totale des investissements prévus pour l' année x + 1

24.2 CAPINV2 INS AN 1 TAUX 324008 Taux d' investissement par rapport à la valeur ajoutée

25 Capital investi CINVEST SIT AN id. ET, id ET 25.1 CINVEST1 INS AN 0 VAUGCAP 324002 Valeur de 1' augmentation du capital  
 VAUGBAN 324003 Valeur de 1' augmentation du capital en biens achetés neufs  
 VAUGBAU 324004 Valeur de 1' augmentation de capital en biens achetés usagés  
 VAUGBP 324005 Valeur de 1' augmentation de capital en biens produits par 1' ET  
 RED 324006 Valeur des réductions de capital

25.2 CINVEST2 INS AN 1 VTINV 324007 Valeur de 1' investissement

25.3 CINVEST3 SANS INEX 0 VPREV 324010 Valeur prévue pour 1' année x + 1  
 DES 324011 Description de 1' investissement prévu pour 1' année x + 1

26 Aide à 1' inv. AINVAA SIT AN id. EP EP 26.1 AINVAI MAE, Min. VAR 0 VTINV 414005 Valeur totale des investissements à aider  
 Fin., Min. NAT 414006 Nature des investissements à aider  
 Clas. Moy., INTNOM 414016 En cas d' aide à 1' investissement de type 1, intérêt nominal demandé à 1' EP  
 SEER, CMGES PINT 414017 En cas d' aide à 1' investissement de type 1, part de 1' intérêt payé par 1' Etat  
 VSUBA 414018 En cas d' aide à 1' investissement de type 1, valeur globale des subside  
 VSUBB 414019 En cas d' aide à 1' investissement de type 2, valeur globale des subside

26.2 AINVAA2 MAE, Min. VAR 0 EPAPP 414025 En cas d' aide à 1' investissement de type 7, identifiant de EP ayant fait 1' apport  
 Fin., SEER NAT 414026 En cas d' aide à 1' investissement de type 7, nature des biens apportés  
 VAL 414027 En cas d' aide à 1' investissement de type 7, valeur des biens apportés



26.3 AINVA3 SANS INEX 0 DES 414007 En cas d' aide à l' investissement pour la création d' un procédé de production, description du proc.  
VAL 414020 En cas d' aide à l' investissement de type 4, valeur du crédit garanti par l' Etat  
CETAT 414021 En cas d' aide à l' investissement de type 4, contribution reçue de l' Etat  
EPBEN 414022 En cas d' aide à l' investissement de type 4, contribution de l' EP bénéficiaire de la garantie  
CORGCR 414023 En cas d' aide à l' investissement de type 4, contribution de l' organisme créancier  
MONT 414024 En cas d' aide à l' investissement de type 4, montant payé par l' Etat pour honorer sa garantie

27.1 AINVA4 Min.Clas. VAR 0 VSUBV 414203 Valeur des subventions en intérêt  
VINV 414204 Valeur des investissements aidés

27.2 AINVA5 SANS INEX 0 CREDIT 414205 Valeur des crédits accordés, garantis par l' Etat  
CONTRIB 414206 Contribution reçue de l' Etat  
GAR 414207 Montant payé par l' Etat pour honorer sa garantie

28 AINVA6 SANS INEX 0 DINT 414209 Date de l' intervention initiale du conseiller

29 AINVA7 MAE,Min. VAR 0 414003 EP ayant demandé une aide à l' investissement  
Fin.,Min. est-elle PME ?  
SEER,CMCES EPPME  
NDOSS 414004 N° dossier demande aide à l' investissement  
ETBEN 414008 ET de EP bénéficiaire de l' aide à l' investissement  
BUT 414009 But de l' investissement  
DMP 414010 Date de mise en place d' un effectif min. suite à l' aide à l' investissement  
NWENT 414011 Nombre min. d' emplois imposés lors de l' aide à l' investissement  
DACH 414012 Date de l' achèvement du programme d' investis.  
NEMPC 414013 Nombre d' emplois créés après le programme d' investissements  
DDEB 414014 Date de début de l' octroi d' aide à l' invest.  
DCLO 414015 Date de clôture de l' octroi d' aide à l' inv.

30.1 NAIDINV1 MAE,Min. VAR 0 414102 Id. de l' ET de EP pour lequel une nouvelle aide à l' investissement est demandée  
Fin.,Min. ETDEM  
Clas. Moy., VAL 414104 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
SEER,CMCES NAT 414105 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
11, nature de l' investissement à aider



BUT 414106 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
11, but de l' investissement à aider  
NEMPL 414107 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
11, nombre d' emplois escomptés  
DACH 414108 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
11, date prévue d' achèvement du progr. d' inv.  
COMMUNE 414109 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
11, commune d' implantation du nouvel inv.  
ZONING 414110 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
11, zoning d' implantation  
CNOM 414118 En cas d' aide à la création d' une EP, nom de la  
nouvelle EP  
CACT 414119 En cas d' aide à la création d' une EP, activité  
principale de la nouvelle EP  
CLOCAL 414120 En cas d' aide à la création d' une EP, localisation  
de l' ET principal  
CNAT 414121 En cas d' aide à la création d' une EP, nature de  
l' investissement  
CVAL 414122 En cas d' aide à la création d' une EP, valeur de  
l' investissement  
CNEMP 414123 En cas d' aide à la création d' une EP, nombre  
d' emplois escomptés  
CFJUR 414124 En cas d' aide à la création d' une EP, forme  
juridique adoptée  
CVCAPINV 414125 En cas d' aide à la création d' une EP, valeur du  
capital initial

30.2 NAIDINV2 SANS INEX 0 ZONE 414111 En cas de nouvelle aide à l' inv. de type 1 à 6 et  
11, zone (plan de secteur) d' implantation

30.3 NAIDINV3 MAE,Min. VAR 0 IDAPP 414112 En cas de nouvelle aide à l' inv. de type 7, identi-  
fiant de l' apporteur  
Fin.,SEER NAT 414113 En cas de nouvelle aide à l' inv. de type 7, nature  
des apports  
NEMPL 414114 En cas de nouvelle aide à l' inv. de type 7, nombre  
d' emplois escomptés

31 Description du DESPRODA SIT AN 1d. ET ET 31.1 DESPROD1 INS AN 0 CONSCAP 323003 Consommation de capital (y compris le loyer du  
processus de matériel acquis par leasing)  
production

31.2 DESPROD2 INS AN 1 PCTVA 323004 Part du coût du travail dans la valeur ajoutée  
PCCVA 323005 Part de la consommation de capital dans la valeur  
ajoutée

32 Description du DESPROD3 EV 1d. ET, ET 32 DESPROD3 SANS INEX 0 DUREE 322004 Durée de la mise en place de la capacité de prod.  
processus de 1d.Prod. PROD CAPAC 322005 Capacité actuelle de production  
production TRANSP 322006 Type de transports utilisés pour les produits  
DEBUT 222003 Date (année) de début de fabrication



33 Description du DESPROD8 SIT AN 1d. EP EP 33.1 DESPROD4 ONSS, AN 1 VIB 323001 Valeur des immobilisations brutes par personne  
processus de RAPPAÑEP VIN 323002 Valeur des immobilisations nettes par personne  
production

33.2 DESPROD5 SANS INEX 0 TAMORT 323006 Taux d'amortissement pratiqué

34 Description du DESPROD6 EV 1d. ET, 1d. ET 34 DESPROD6 SANS INEX 0 DES 322003 Description de cette expérience  
processus de Tch. Prod. TECHNOL DDES 322002 Date de cette description  
production

35 Caract. de la CARPHA SIT TRIM 1d. ET ET 35.1 CARPH1 ONSS TRIM 0 NBTRAV 311001 Nombre de travailleurs occupés  
réserve de NBMAN 311002 Nombre de manuels occupés  
potentiel NBINT 311003 Nombre d'intellectuels occupés  
humain NBHOM 311005 Nombre d'hommes occupés  
NBFEF 311006 Nombre de femmes occupées  
NBMVU 311007 Nombre de personnes de moins de 21 ans occupées  
NBEVUC 311008 Nombre de personnes de 21 à 50 ans occupées  
NVPC 311009 Nombre de personnes de plus de 50 ans occupées

35.2 CARPH2 INS DEC 0 TSYND 311014 Taux de syndicalisation de la main-d'oeuvre

36 Caract. de la CARPH3 SIT TRIM 1d. ET, 1d. ET 36 CARPH3 INS AN 0 NBPNA 311004 Personnel non assujéti à la sécurité sociale des  
réserve de S. Activ. SECTACT travailleurs  
pot. humain NBTRAV 311011 Nombre de travailleurs occupés  
NBMAN 311012 Nombre de manuels occupés  
NBINT 311013 Nombre d'intellectuels occupés

37 Caract. de la CARPH4 SIT TRIM 1d. ET, 1d. ET 37 CARPH4 SANS INEX 0 NBPEKS 311010 Nombre de personnes occupées ayant une qualification  
réserve de Q. Scol. QSCOL scolaire Q<sub>1</sub>  
pot. humain

38 Caract. de la CARPHB EX 1d. Trav. TRAV 38.1 CARPH5 INAMI VAR 0 NOM 311101 Nom de l'ET/EP qui l'occupe  
réserve de Occup. ADR 311107 Adresse  
pot. humain

38.2 CARPH6 RN SANS 0 ID 311102 Identité  
DNAISS 311103 Date de naissance  
SEXE 311104 Sexe  
NAT 311105 Nationalité  
ETATC 311106 Etat civil

38.3 CARPH7 INS DEC 0 QSCOL 311109 Qualification scolaire (Q<sub>1</sub>)  
QEMPL 311110 Qualification de l'emploi occupé (P<sub>1</sub>)

| 38.4 CHARPH8 SANS                                                 |                                                    |                                 | INEX 0 TTRAV QDEMP | 311108 Type de travail<br>301111 Qualification des 5 derniers emplois                                   |
|-------------------------------------------------------------------|----------------------------------------------------|---------------------------------|--------------------|---------------------------------------------------------------------------------------------------------|
| 39 Charges d' util. CHARPHA SIT TRIM 1d. ET ET                    | 39.1 CHARPH1 ONSS                                  | TRIM 0 VCPTM                    |                    | 312001 Valeur de la cotisation patronale pour les tra-<br>vailleurs manuels                             |
|                                                                   |                                                    |                                 |                    | 312002 Valeur de la cotisation patronale pour les tra-<br>vailleurs intellectuels                       |
|                                                                   |                                                    |                                 |                    | 312003 Valeur de la cotisation salariale pour les tra-<br>vailleurs manuels                             |
|                                                                   |                                                    |                                 |                    | 312004 Valeur de la cotisation salariale pour les tra-<br>vailleurs intellectuels                       |
|                                                                   |                                                    |                                 |                    | 312006 Valeur globale des rémunérations brutes annuelles                                                |
|                                                                   |                                                    |                                 |                    | 312007 Valeur globale des rémunérations brutes annuelles<br>du personnel ouvrier                        |
|                                                                   |                                                    |                                 |                    | 312008 Valeur globale des rémunérations brutes annuelles<br>du personnel employé                        |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
| 39.2 CHARPH2 INS                                                  | AN 0 MGD                                           | MADP                            |                    | 312005 Montant global des dépenses en personnel                                                         |
|                                                                   |                                                    |                                 |                    | 312013 Montant des autres dépenses en personnel                                                         |
| 39.3 CHARPH3 SANS                                                 | INEX 0 VRPNA                                       | SYREM                           |                    | 312009 Valeur globale des rémunérations brutes annuelles<br>du pers. non assujéti à la sécurité sociale |
|                                                                   |                                                    |                                 |                    | 312012 Système de rémunération habituel (R <sub>1</sub> )                                               |
| 40 Charges d' util. CHARPH4 SIT TRIM 1d. ET, 1d ET Q. Empl. QEMPL | 40 CHARPH4 SANS                                    | INEX 0 SALAB SUPPL              |                    | 312010 Salaire annuel brut                                                                              |
|                                                                   |                                                    |                                 |                    | 312011 Supplément par année d' ancienneté                                                               |
| 41 Charges d' util. CHARPHB SIT TRIM 1d. Trav. Occupé             | 41.1 CHARPH5 ONSS                                  | TRIM 0 CONTRAT REMBA NBJR REGCJ |                    | 312101 Type de contrat                                                                                  |
|                                                                   |                                                    |                                 |                    | 312104 Rémunération brute annuelle                                                                      |
|                                                                   |                                                    |                                 |                    | 312105 Nombre de journées rémunérées pour un trimestre                                                  |
|                                                                   |                                                    |                                 |                    | 312113 Est-il soumis à un régime de 5 jours de travail ?                                                |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
| 41.2 CHARPH6 SANS                                                 | INEX 0 DACONT DUCONT HR HRN HRPE HRDE HRTD HRWE CH |                                 |                    | 312102 Date de conclusion du contrat                                                                    |
|                                                                   |                                                    |                                 |                    | 312103 Durée du contrat                                                                                 |
|                                                                   |                                                    |                                 |                    | 312106 Nombre d' heures de travail rémunérées pour 1 trim.                                              |
|                                                                   |                                                    |                                 |                    | 312107 Nombre d' heures de travail rémunérées pour 1 trim.<br>en heures normales                        |
|                                                                   |                                                    |                                 |                    | 312108 Nombre d' heures de travail rémunérées pour 1 trim.<br>en heures 1 équipe                        |
|                                                                   |                                                    |                                 |                    | 312109 Nombre d' heures de travail rémunérées pour 1 trim.<br>en heures 2 équipes                       |
|                                                                   |                                                    |                                 |                    | 312110 Nombre d' heures de travail rémunérées pour 1 trim.<br>en heures 3 équipes                       |
|                                                                   |                                                    |                                 |                    | 312111 Nombre d' heures de travail rémunérées pour 1 trim.<br>en heures week-end                        |
|                                                                   |                                                    |                                 |                    | 312112 Nombre d' heures chômées sur un trimestre                                                        |
|                                                                   |                                                    |                                 |                    |                                                                                                         |
|                                                                   |                                                    |                                 |                    |                                                                                                         |



|                                                   |         |        |                      |              |      |                            |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------------|---------|--------|----------------------|--------------|------|----------------------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 42 Amélioration de la réserve de potentiel humain | AMELPH1 | SIT AN | 1d. ET               | ET           | 42   | AMELPH1 SANS               | INEX 0 SFPER<br>PREC        | 315001 Somme globale consacrée à la formation permanente<br>315002 Nombre de personnes ayant suivi des cours de recyclage<br>315003 Nombre de personnes (cadres) ayant suivi des cours de recyclage<br>315004 Nombre de personnes (employés) ayant suivi des cours de recyclage<br>315005 Nombre de personnes (ouvriers) ayant suivi des cours de recyclage                                                                                      |
|                                                   | AMELPH2 | EV     | 1d. Trav.            | TRAV         | 43   | AMELPH2 SANS               | INEX 0 HFORM                | 315101 Nombre d'heures de formation suivies pour le recyclage<br>315102 Nombre de crédits d'heures-semaine<br>315103 Nombre d'heures de formation sur crédits-d'heures suivie<br>315104 Nom du centre de formation où la formation est suivie<br>315105 Date d'entrée dans un centre de formation perm.<br>315106 Qualification complémentaire obtenue après une formation permanente<br>315107 Date d'obtention de la qualification complément. |
|                                                   | RENP1   | EV     | id. ET,<br>id. Empl. | ET<br>EMPOFF | 44   | RENP1, ONEM,Org.<br>Presse | VAR 0 DVAC<br>DES<br>SOURCE | 311202 Date de la vacance<br>311203 Description de l'offre<br>311204 Source de l'information                                                                                                                                                                                                                                                                                                                                                     |
|                                                   | RENP1   | SIT AN | 1d. ET               | ET           | 45.1 | RENP2                      | ONEM,Org.<br>Presse         | 311201 Nombre d'emplois offerts                                                                                                                                                                                                                                                                                                                                                                                                                  |
|                                                   | RENP3   | SANS   |                      |              | 45.2 | RENP3                      | INEX 0 NBEONO<br>QEMPL      | 311205 Nombre d'emplois offerts non occupés<br>311206 Qualification de l'emploi non occupé                                                                                                                                                                                                                                                                                                                                                       |
| 46 Activité de production                         | ACTPROD | SIT AN | 1d. ET               | ET           | 46   | ACTPROD INS                | AN 1 VAPM<br>VACF<br>DEGTR  | 210002 Valeur ajoutée brute au prix du marché (TVA exclue)<br>210003 Valeur ajoutée brute au coût des facteurs (hors TVA)<br>210004 Degré de transformation des produits entrants                                                                                                                                                                                                                                                                |
|                                                   | ACHCESS | SIT AN | 1d. ET               | ET           | 47.1 | ACHCESS1 INS               | AN 0 VAPE<br>VAPR<br>VMPR   | 211001 Valeur des achats de produits entrants (hors TVA déductible)<br>211005 Valeur des achats de produits revendus comme tels (hors TVA déductible)<br>211012 Valeur des matières et produits reçus d'autres ET de l'EP (hors TVA déductible)                                                                                                                                                                                                  |

47.2 ACCESS2 SANS INEX 0 VAMR 211002 Valeur des achats sur le marché régional (hors TVA déductible)  
 VAMB 211003 Valeur des achats sur le marché belge (hors TVA déductible)  
 VAME 211004 Valeur des achats sur le marché étranger (hors TVA déductible)  
 QAPE 211006 Quantité d'achats de produits entrants  
 QAMR 211007 Quantité d'achats sur le marché régional  
 QAMB 211008 Quantité d'achats sur le marché belge  
 QAME 211009 Quantité d'achats sur le marché étranger  
 REGION 211010 Région d'origine des achats effectués sur le marché belge  
 PAYS 211011 Pays d'origine des importations

| 48 Stocks | STOCKS | SIT AN | 1d. ET | ET | 48.1 STOCKS1 | INS | AN | 0      | VSDE<br>VSPED                                                                                                                                                                                                          | 214001 Valeur des stocks en début d'exercice (TVA excl.)<br>214002 Valeur des stocks et produits entrants (début ex.)<br>(TVA exclue) |
|-----------|--------|--------|--------|----|--------------|-----|----|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
|           |        |        |        |    |              |     |    | VSPECD | 214003 Valeur des stocks de produits en-cours (début ex.)<br>214004 Valeur des stocks de produits sortants (début ex.)<br>214005 Valeur des stocks de produits revendus comme tels (début ex.) (TVA déductible exclue) |                                                                                                                                       |
|           |        |        |        |    |              |     |    | QSPSD  | 214009 Quantité des stocks de produits sortants (deb. ex.)<br>214010 Valeur des stocks en fin d'exercice (TVA exclue)<br>214011 Valeur des stocks de produits entrants (fin ex.)<br>(hors TVA déductible)              |                                                                                                                                       |
|           |        |        |        |    |              |     |    | VSPECF | 214012 Valeur des stocks de produits en-cours (fin ex.)<br>214013 Valeur des stocks de produits sortants (fin ex.)<br>214014 Valeur des stocks de produits revendus comme tels (fin ex.) (TVA déductible exclue)       |                                                                                                                                       |
|           |        |        |        |    |              |     |    | QSPSF  | 214018 Quantité des stocks de produits sortants (fin ex.)                                                                                                                                                              |                                                                                                                                       |

48.2 STOCKS2 SANS INEX 0 QSDE 214006 Quantité des stocks en début d'exercice  
 QSPED 214007 Quantité des stocks de produits entrants (deb. ex.)  
 VSPECD 214008 Quantité des stocks de produits en-cours (deb. ex.)  
 QSFE 214015 Quantité des stocks en fin d'exercice  
 QSPF 214016 Quantité des stocks de produits entrants (fin ex.)  
 QSPCF 214017 Quantité des stocks de produits en-cours (fin ex.)

| 49 Consommation | CONSOM1 | SIT AN | 1d. ET, 1d. ET<br>Mat. Prem. MP | ET | 49 | CONSOM1 | INS | AN | 0 | QCONS<br>VCONS | 212002 Quantité consommée<br>212003 Valeur consommée (hors TVA) |
|-----------------|---------|--------|---------------------------------|----|----|---------|-----|----|---|----------------|-----------------------------------------------------------------|
|-----------------|---------|--------|---------------------------------|----|----|---------|-----|----|---|----------------|-----------------------------------------------------------------|

| 50 Consommation | CONSOM2 | SIT AN | 1d. ET, 1d. ET<br>Sour. Energy SEN | ET | 50 | CONSOM2 | INS | AN | 0 | QCONS<br>VCONS | 212005 Quantité consommée<br>212006 Valeur consommée (hors TVA) |
|-----------------|---------|--------|------------------------------------|----|----|---------|-----|----|---|----------------|-----------------------------------------------------------------|
|-----------------|---------|--------|------------------------------------|----|----|---------|-----|----|---|----------------|-----------------------------------------------------------------|



| 51 Consommation | CONSOM | SIT AN | 1d. ET | ET | 51.1 CONSOM3 | INS | AN | 0 VGAC | 212007 Valeur globale des autres consommations (hors TVA)                             |
|-----------------|--------|--------|--------|----|--------------|-----|----|--------|---------------------------------------------------------------------------------------|
|                 |        |        |        |    |              |     |    | VCE    | 212008 Valeur de la consommation en eau                                               |
|                 |        |        |        |    |              |     |    | VCER   | 212009 Valeur de la consommation en emballages et réci-<br>pients (hors TVA)          |
|                 |        |        |        |    |              |     |    | VCMER  | 212010 Valeur de la consommation en matériel d'entretien<br>et réparations (hors TVA) |
|                 |        |        |        |    |              |     |    | VCPO   | 212011 Valeur de la consommation en petits outils..., non<br>dénommés ailleurs        |
|                 |        |        |        |    |              |     |    | QGAC   | 212012 Quantité globale des autres consommations                                      |
|                 |        |        |        |    |              |     |    | OCE    | 212013 Quantité de la consommation en eau                                             |
|                 |        |        |        |    |              |     |    | VFT    | 212015 Valeur des frais de transports effectués par des<br>tiers (hors TVA)           |
|                 |        |        |        |    |              |     |    | VTST   | 212016 Valeur des travaux donnés en sous-traitance (hors<br>TVA)                      |
|                 |        |        |        |    |              |     |    | VCHFD  | 212017 Valeur des charges diverses, frais de déplacement<br>du personnel (hors TVA)   |
|                 |        |        |        |    |              |     |    | VDER   | 212018 Valeur des dépenses d'entretien et de réparation<br>(TVA déductible exclue)    |
|                 |        |        |        |    |              |     |    | VPAV   | 212019 Valeur des primes d'assurance incendie, vol vol-<br>tures, etc....             |
|                 |        |        |        |    |              |     |    | VLBFP  | 212020 Valeur des loyers des bâtiments utilisés à des fins<br>professionnelles        |
|                 |        |        |        |    |              |     |    | VLBEQ  | 212021 Valeur des loyers de biens d'équipement (TVA dé-<br>ductible exclue)           |
|                 |        |        |        |    |              |     |    | VFB    | 212022 Valeur des frais bancaires (TVA déductible exclue)                             |

|              |      |              |                                                                                |
|--------------|------|--------------|--------------------------------------------------------------------------------|
| 51.2 CONSOM4 | SANS | INEX 0 QCMER | 212014 Quantité de la consommation en matériel d'entretien<br>et de réparation |
|              |      | SSTR         | 212023 Identifiant des ET/EP où se fait la sous-traitance                      |

| 52 Production | PRODUA | SIT AN | 1d. ET | ET | 52.1 PROD1 | SANS         | INEX 0 LPF<br>LPS | 222001 Liste des produits fournis<br>222002 Liste des produits supprimés au cours de 1 <sup>e</sup> année |
|---------------|--------|--------|--------|----|------------|--------------|-------------------|-----------------------------------------------------------------------------------------------------------|
|               |        |        |        |    | 52.2 PROD2 | INS,<br>ONSS | AN 1 INDPH        | 313002 Indice de production moyenne                                                                       |
|               |        |        |        |    | 52.3 PROD3 | INS          | AN 0 VTLM         | 215006 Valeur des travaux effectués pour lui-même (TVA ex.)                                               |

| 53 Production | PRODUB | SIT AN | 1d. ET,<br>1d.Prod. | ET<br>PROD | 53.1 PROD4 | INS | AN 0 QFLM | 215001 Quantité fabriquée pour l' ET lui-même               |
|---------------|--------|--------|---------------------|------------|------------|-----|-----------|-------------------------------------------------------------|
|               |        |        |                     |            |            |     | QFT       | 215002 Quantité fabriquée pour des tiers                    |
|               |        |        |                     |            |            |     | QFA       | 215003 Quantité fabriquée dans d' autres ET pour son compte |
|               |        |        |                     |            |            |     | QC        | 215004 Quantité cédée à d' autres ET de l' EP               |
|               |        |        |                     |            |            |     | QPFC      | 215005 Quantité de produits finis consommés dans l' ET      |
|               |        |        |                     |            | 53.2 PROD5 | INS | AN 1 EPQP | 313003 Estimation du prix de la quantité produite           |
|               |        |        |                     |            |            |     | VQP       | 313004 Valeur de la quantité produite                       |



| 54 Ventes<br>(Volume)                           | VENTESV  | SIT AN | 1d. ET, 1d<br>Prod.     | ET<br>PROD   | 54.1 VENTESV1 INS              | AN     | 0 VVTC             | 213001 Valeur des ventes ou travaux au cours de l' ex-<br>ercice (hors TVA)                                                                          |
|-------------------------------------------------|----------|--------|-------------------------|--------------|--------------------------------|--------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                 |          |        |                         |              |                                |        | VUDE               | 213004 Valeur des ventes destinées à l' étranger                                                                                                     |
|                                                 |          |        |                         |              |                                |        | QVEX               | 213006 Quantités vendues au cours de l' exercice                                                                                                     |
|                                                 |          |        |                         |              |                                |        | QVE                | 213009 Quantités vendues à l' étranger                                                                                                               |
| -----                                           |          |        |                         |              |                                |        |                    |                                                                                                                                                      |
|                                                 |          |        |                         |              | 54.2 VENTESV2 SANS             | INEX 0 | VVR                | 213002 Valeur des ventes destinées à la région (TVA excl.)                                                                                           |
|                                                 |          |        |                         |              |                                |        | VVAR               | 213003 Valeur des ventes destinées aux autres régions (hors<br>TVA)                                                                                  |
|                                                 |          |        |                         |              |                                |        | QVR                | 213007 Quantités vendues dans la région                                                                                                              |
|                                                 |          |        |                         |              |                                |        | QVAR               | 213008 Quantités vendues dans les autres régions                                                                                                     |
|                                                 |          |        |                         |              |                                |        | REGION             | 213010 Région de destination des ventes                                                                                                              |
|                                                 |          |        |                         |              |                                |        | PAYS               | 213011 Pays de destination des ventes                                                                                                                |
| -----                                           |          |        |                         |              |                                |        |                    |                                                                                                                                                      |
| 55 Ventes<br>(Volume)                           | VENTESV3 | SIT AN | 1d. ET                  | ET           | 55 VENTESV3 INS                | AN     | 0 VVPST            | 213005 Valeur des ventes de produits sortis sans trans-<br>formation (hors TVA)                                                                      |
|                                                 |          |        |                         |              |                                |        | VPCA               | 213014 Valeur des produits cédés à d' autres ET de l' EP                                                                                             |
|                                                 |          |        |                         |              |                                |        | VTET               | 213015 Valeur des travaux effectués pour des tiers (TVA ex.)                                                                                         |
|                                                 |          |        |                         |              |                                |        | VPA                | 213016 Valeur des produits accessoires                                                                                                               |
| -----                                           |          |        |                         |              |                                |        |                    |                                                                                                                                                      |
| 56 Pollution                                    | POLL1    | EV     | 1d. ET                  | ET           | 56 POLL1 SANS                  | INEX 0 | NBPLAIN<br>AMENDES | 240003 Nombre de plaintes contre la pollution<br>240004 Montant des amendes et indemnités versées pour<br>nuisances                                  |
| -----                                           |          |        |                         |              |                                |        |                    |                                                                                                                                                      |
| 57 Pollution                                    | POLL2    | EV     | 1d. ET, 1d<br>Ty. Poll. | ET<br>TYPOLL | 57 POLL2 SANS                  | INEX 0 | MESURE             | 240002 Mesure de la pollution                                                                                                                        |
| -----                                           |          |        |                         |              |                                |        |                    |                                                                                                                                                      |
| 58 Pollution                                    | POLL3    | EV     | 1d. EP<br>Pilote        | EPILL        | 58 POLL3 MIn.Santé<br>Publique | VAR    | 0 PETAT            | 240007 Part prise par l' Etat dans le financement des équi-<br>pements de dépollution                                                                |
| -----                                           |          |        |                         |              |                                |        |                    |                                                                                                                                                      |
| 59 Pollution                                    | POLL4    | EV     | 1d. Typ.<br>Poll.       | TYPOLL       | 59 POLL4 MIn.Santé<br>Publique | VAR    | 0 MAIDES           | 240005 Montant des aides disponibles de l' Etat, pour les<br>équipements de dépollution                                                              |
|                                                 |          |        |                         |              |                                |        | LOCEPU             | 240006 Localisation des épurateurs publics                                                                                                           |
|                                                 |          |        |                         |              |                                |        | RECH               | 240008 Identifiant des recherches entreprises à la deman-<br>de de l' Etat                                                                           |
| -----                                           |          |        |                         |              |                                |        |                    |                                                                                                                                                      |
| 60 Structure du<br>prix de<br>revient<br>global | STRPRG   | SIT AN | 1d. ET                  | ET           | 60 STRPRG MAE                  | VAR    | 0 PREV<br>PPRMP    | 221101 Prix de revient pour l' ensemble de la production<br>221102 % du prix de revient pour l' ensemble de la produc-<br>tion en matières premières |
|                                                 |          |        |                         |              |                                |        | PPRMAC             | 221103 % du prix de revient pour l' ensemble de la produc-<br>tion en mat. accessoires et de conditionnement                                         |
|                                                 |          |        |                         |              |                                |        | PPRSACH            | 221104 % du prix de revient pour l' ensemble de la produc-<br>tion en salaires, appoint, et charges sociales                                         |
|                                                 |          |        |                         |              |                                |        | PPREN              | 221105 % du prix de revient pour l' ensemble de la produc-<br>tion en énergie                                                                        |
|                                                 |          |        |                         |              |                                |        | PPRAM              | 221106 % du prix de revient pour l' ensemble de la produc-<br>tion en amortissements                                                                 |



PPRFG 221107 % du prix de revient pour l' ensemble de la produc-  
tion en frais généraux (dont frais financiers)  
PPRFV 221108 % du prix de revient pour l' ensemble de la produc-  
tion en frais de vente  
PPRDIV 221109 % du prix de revient pour l' ensemble de la produc-  
tion en divers

| 61 Structure du<br>prix de<br>revient par<br>produit | STRPRPRO SIT AN | 1d. ET, 1d<br>Prod | ET<br>PROD | 61 | STRPRPRO MAE | VAR | 0 | PVEU<br>PREV<br>VMP<br>VMAC<br>VSACH<br>VEN<br>VAM<br>VFG<br>VJV<br>VDIV<br>MBEN<br>DAPPL | Prix de vente ex-usine<br>Prix de revient<br>Valeur dans le prix de revient des matières prem.<br>Valeur dans le prix de revient des matières acces-<br>soires et de conditionnement<br>Valeur dans le prix de revient des salaires, ap-<br>pointements et charges sociales<br>Valeur dans le prix de revient de l' énergie<br>Valeur dans le prix de revient des amortissements<br>Valeur dans le prix de revient des frais généraux<br>(dont frais financiers)<br>Valeur dans le prix de revient des frais de vente<br>Valeur dans le prix de revient des divers<br>Marge bénéficiaire du producteur<br>Date d' application du prix de vente |
|------------------------------------------------------|-----------------|--------------------|------------|----|--------------|-----|---|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|------------------------------------------------------|-----------------|--------------------|------------|----|--------------|-----|---|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 62 Productivité | PRODUCTI SIT AN | 1d. ET | ET | 62 | PRODUCTI<br>ONSS | AN | 1 | PMVO<br>PMVA<br>DEPTR<br>PRM<br>IMPR<br>VAMT<br>IVAMT | Productivité moyenne de la main-d'oeuvre (en volume)<br>Productivité moyenne de la main-d'oeuvre (en valeur)<br>Dépense brute moyenne par travailleur<br>Productivité moyenne par unité homme-dépense<br>Indice de productivité moyenne par unité<br>homme-dépense<br>Valeur ajoutée moyenne par travailleur<br>Indice de valeur ajoutée moyenne par travailleur |
|-----------------|-----------------|--------|----|----|------------------|----|---|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|-----------------|-----------------|--------|----|----|------------------|----|---|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 63 Résultats d'<br>exploitation | RESEXA | SIT AN | 1d. EP | EP | 63.1 RESEX1 | RAPPANNEP AN | 1 | VCFN<br>VAF<br>VCAF<br>VPB | Valeur du cash flow net de l' exercice<br>Valeur de l' autofinancement<br>Valeur de la capacité d' autofinancement<br>Valeur du profit brut courant avant impôt sur les<br>bénéfices |
|---------------------------------|--------|--------|--------|----|-------------|--------------|---|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|---------------------------------|--------|--------|--------|----|-------------|--------------|---|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 63.2 RESEX2 | Min. Fin. AN | 0 | VCHA<br>VREX<br>VPP<br>VFEX | Valeur du chiffre d' affaires de l' exercice<br>Valeur du résultat d' exploitation de l' exercice<br>Valeur des provisions et prévisions de l' exercice<br>Valeur des frais d' exploitation de l' exercice |
|-------------|--------------|---|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|-------------|--------------|---|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| 63.3 RESEX3 | SANS | INEX | 0 | VPFIN<br>VCHF<br>VAM<br>VREP | Valeur des produits financiers de l' exercice<br>Valeur des charges financières de l' exercice<br>Valeur des amortissements de l' exercice<br>Valeur du report de l' exercice précédent |
|-------------|------|------|---|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|-------------|------|------|---|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## 64 Résultats d'exploitation

RESEXB SIT AN 1d. ET ET 64.1 RESEX4 INS AN 0 VSUBV 423006 Valeur des subventions d'exploitations reçues

64.2 RESEX5 INS AN 1 VEXCB 423007 Valeur de l'excédent brut d'exploitation

64.3 RESEX6 SANS INEX 0 VPRIMES 324012 Valeur des primes d'assurance incendie  
MASS 324013 Montants assurés contre l'incendie

## 65 Distribution du bénéfice

65.1 DISBEN1 RAPPANNEP AN 0 VDIV 424001 Valeur des dividendes  
VTANT 424002 Valeur des tantièmes  
VDONS 424003 Valeur des dons  
VBEN 424004 Valeur du bénéfice reporté65.2 DISBEN2 ONSS, MAE AN 1 VMS 424007 Valeur de la masse salariale, à prix constants  
IEVMS 424008 Indice d'évolution de la masse salariale, à prix constants65.3 DISBEN3 RAPPANNEP AN 1 VDCV 424005 Val. des dividendes, compte tenu du coût de la vie  
IEVOL 424006 Indice d'évolution des dividendes, à prix constants

## 66 Distribution du bénéfice

66 DISBEN4 SIT AN 1d. ET ET 66 DISBEN4 INS AN 1 PSVA 424011 Part (%) des salaires dans la valeur ajoutée  
PPVA 424012 Part (%) du profit dans la valeur ajoutée

## 67 Indice des prix de détail

67 INDP SIT MOIS 67 INDP MAE MOIS 0 IPD 111101 Indice des prix de détail

## 68 Imposition

68 IMPOS SIT AN 1d. ET ET 68 IMPOS INS-SPA AN 0 MDACH 432009 Montant annuel global de la TVA déductible sur les achats de combustibles, de matières premières  
VTVAE 432011 Valeur de la TVA versée à l'Etat  
VAII 432012 Valeur des autres impôts indirects  
VTAV 432010 Valeur annuelle de la TVA perçue sur les ventes



ANNEXE 2 :

ENSEMBLE DES DESCRIPTIFS DES  
=====

RELATIONS DE LA BASE DE DONNEES  
=====

DESCRIPTIF GENERAL DES RELATIONS : REL

-----

|                                  |        |
|----------------------------------|--------|
| IDET1 8 FIDET1 DBER A1 144       | (1.1)  |
| IDET2 2 FIDET2 DBER A1 21        | (1.2)  |
| IDEP 9 FIDEP DBER A1 180         | (2)    |
| DONSS 7 FDETONSS DBER A1 27      | (3.1)  |
| DPP 7 FDETPP DBER A1 27          | (3.2)  |
| DIS 7 FDETIS DBER A1 27          | (3.3)  |
| LICENC 4 FLIC DBER A1 27         | (4)    |
| CAPASSIF 4 FCAPASS DBER A1 23    | (5)    |
| CAPP1 11 FCAPP1 DBER A1 93       | (6.1)  |
| CAPP2 8 FCAPP2 DBER A1 43        | (6.2)  |
| CAPP3 6 FCAPP3 DBER A1 43        | (6.3)  |
| CAPP4 5 FCAPP4 DBER A1 25        | (6.4)  |
| CAPP5 7 FCAPP5 DBER A1 27        | (7)    |
| CAPP6 8 FCAPP6 DBER A1 47        | (8)    |
| CAPP7 6 FCAPP7 DBER A1 28        | (9)    |
| CAPP8 4 FCAPP8 DBER A1 23        | (10)   |
| CAPE1 5 FCAPE1 DBER A1 33        | (11.1) |
| CAPE2 7 FCAPE1 DBER A1 63        | (11.2) |
| CAPE3 13 FCAPE3 DBER A1 53       | (12)   |
| CAPE4 14 FCAPE4 DBER A1 77       | (13)   |
| CAPACTIF 4 FCAPACT DBER A1 23    | (14)   |
| CAPANIMM1 7 FCAPANI1 DBER A1 53  | (15.1) |
| CAPANIMM2 5 FCAPANI2 DBER A1 33  | (15.2) |
| CAPAIMM1 9 FCAPAIM1 DBER A1 74   | (16)   |
| CAPAIMM2 6 FCAPAIM2 DBER A1 28   | (17)   |
| CAPHYS1 10 FCAPHYS1 DBER A1 69   | (18)   |
| CAPHYS2 6 FCAPHYS2 DBER A1 21    | (19)   |
| CAPHYS3 6 FCAPHYS3 DBER A1 22    | (20)   |
| CAPHYS4 6 FCAPHYS4 DBER A1 21    | (21)   |
| CAPHYS5 6 FCAPHYS5 DBER A1 22    | (22)   |
| RATFIN1 11 FRATFIN1 DBER A1 53   | (23.1) |
| RATFIN2 5 FRATFIN2 DBER A1 27    | (23.2) |
| RATFIN3 4 FRATFIN3 DBER A1 19    | (23.3) |
| RATFIN4 8 FRATFIN4 DBER A1 43    | (23.4) |
| CAPINV1 5 FCAPINV1 DBER A1 33    | (24.1) |
| CAPINV2 4 FCAPINV2 DBER A1 17    | (24.2) |
| CINVEST1 10 FCINVES1 DBER A1 65  | (25.1) |
| CINVEST2 6 FCINVES2 DBER A1 25   | (25.2) |
| CINVEST3 7 FCINVES3 DBER A1 65   | (25.3) |
| AINVA1 9 FAIDINV1 DBER A1 71     | (26.1) |
| AINVA2 6 FAIDINV2 DBER A1 54     | (26.2) |
| AINVA3 9 FAIDINV3 DBER A1 103    | (26.3) |
| AINVA4 5 FAIDINV4 DBER A1 33     | (27.1) |
| AINVA5 6 FAIDINV5 DBER A1 43     | (27.2) |
| AINVA6 8 FAIDINV6 DBER A1 27     | (28)   |
| AINVA7 17 FAIDINV7 DBER A1 117   | (29)   |
| NAIDINV1 24 FNOUVAI1 DBER A1 276 | (30.1) |
| NAIDINV2 8 FNOUVAI2 DBER A1 37   | (30.2) |
| NAIDINV3 10 FNOUVAI3 DBER A1 63  | (30.3) |
| DESPROD1 8 FDESPRO1 DBER A1 54   | (31.1) |
| DESPROD2 5 FDESPRO2 DBER A1 21   | (31.2) |
| DESPROD3 6 FDESPRO3 DBER A1 15   | (32)   |
| DESPROD4 5 FDESPRO4 DBER A1 33   | (33.1) |
| DESPROD5 4 FDESPRO5 DBER A1 17   | (33.2) |
| DESPROD6 6 FDESPRO6 DBER A1 93   | (34)   |



|          |    |          |      |    |     |        |
|----------|----|----------|------|----|-----|--------|
| CARPH1   | 13 | FCAPOHU1 | DBER | A1 | 63  | (35.1) |
| CARPH2   | 6  | FCAPOHU2 | DBER | A1 | 19  | (35.2) |
| CARPH3   | 11 | FCAPOHU3 | DBER | A1 | 43  | (36)   |
| CARPH4   | 8  | FCAPOHU4 | DBER | A1 | 24  | (37)   |
| CARPH5   | 3  | FCAPOHU5 | DBER | A1 | 106 | (38.1) |
| CARPH6   | 6  | FCAPOHU6 | DBER | A1 | 68  | (38.2) |
| CARPH7   | 3  | FCAPOHU7 | DBER | A1 | 12  | (38.3) |
| CARPH8   | 3  | FCAPOHU8 | DBER | A1 | 36  | (38.4) |
| CHARPH1  | 12 | FCHARPH1 | DBER | A1 | 85  | (39.1) |
| CHARPH2  | 7  | FCHARPH2 | DBER | A1 | 35  | (39.2) |
| CHARPH3  | 7  | FCHARPH3 | DBER | A1 | 55  | (39.3) |
| CHARPH4  | 9  | FCHARPH4 | DBER | A1 | 38  | (40)   |
| CHARPH5  | 9  | FCHARPH5 | DBER | A1 | 46  | (41.1) |
| CHARPH6  | 13 | FCHARPH6 | DBER | A1 | 41  | (41.2) |
| AMELPH1  | 8  | FAMELPH1 | DBER | A1 | 47  | (42)   |
| AMELPH2  | 10 | FAMELPH2 | DBER | A1 | 69  | (43)   |
| RENPH1   | 12 | FRENPH1  | DBER | A1 | 92  | (44)   |
| RENPH2   | 4  | FRENPH2  | DBER | A1 | 19  | (45.1) |
| RENPH3   | 5  | FRENPH3  | DBER | A1 | 22  | (45.2) |
| ACTPROD  | 6  | FACTPROD | DBER | A1 | 43  | (46)   |
| ACHCESS1 | 6  | FACHCES1 | DBER | A1 | 43  | (47.1) |
| ACHCESS2 | 12 | FACHCES2 | DBER | A1 | 115 | (47.2) |
| STOCKS1  | 15 | FSTOCKS1 | DBER | A1 | 129 | (48.1) |
| STOCKS2  | 9  | FSTOCKS2 | DBER | A1 | 61  | (48.2) |
| CONSOM1  | 7  | FCONSOM1 | DBER | A1 | 34  | (49)   |
| CONSOM2  | 7  | FCONSOM2 | DBER | A1 | 34  | (50)   |
| CONSOM3  | 18 | FCONSOM3 | DBER | A1 | 159 | (51.1) |
| CONSOM4  | 5  | FCONSOM4 | DBER | A1 | 54  | (51.2) |
| PROD1    | 5  | FPRODUC1 | DBER | A1 | 83  | (52.1) |
| PROD2    | 4  | FPRODUC2 | DBER | A1 | 17  | (52.2) |
| PROD3    | 4  | FPRODUC3 | DBER | A1 | 23  | (52.3) |
| PROD4    | 10 | FPRODUC4 | DBER | A1 | 57  | (53.1) |
| PROD5    | 7  | FPRODUC5 | DBER | A1 | 37  | (53.2) |
| VENTESV1 | 9  | FVENTES1 | DBER | A1 | 53  | (54.1) |
| VENTESV2 | 11 | FVENTES2 | DBER | A1 | 93  | (54.2) |
| VENTESV3 | 7  | FVENTES3 | DBER | A1 | 53  | (55)   |
| POLL1    | 7  | FPOLL1   | DBER | A1 | 32  | (56)   |
| POLL2    | 8  | FPOLL2   | DBER | A1 | 39  | (57)   |
| POLL3    | 6  | FPOLL3   | DBER | A1 | 20  | (58)   |
| POLL4    | 8  | FPOLL4   | DBER | A1 | 109 | (59)   |
| STRPRG   | 12 | FPOLL5   | DBER | A1 | 55  | (60)   |
| STRPRPRO | 17 | FSTRUPRP | DBER | A1 | 137 | (61)   |
| PRODUCTI | 10 | FPRODUCT | DBER | A1 | 71  | (62)   |
| RESEX1   | 7  | FRESEXP1 | DBER | A1 | 53  | (63.1) |
| RESEX2   | 7  | FRESEXP2 | DBER | A1 | 53  | (63.2) |
| RESEX3   | 7  | FRESEXP3 | DBER | A1 | 53  | (63.3) |
| RESEX4   | 4  | FRESEXP4 | DBER | A1 | 23  | (64.1) |
| RESEX5   | 4  | FRESEXP5 | DBER | A1 | 23  | (64.2) |
| RESEX6   | 5  | FRESEXP6 | DBER | A1 | 23  | (64.3) |
| DISBEN1  | 7  | FDISBEN1 | DBER | A1 | 53  | (65.1) |
| DISBEN2  | 5  | FDISBEN2 | DBER | A1 | 29  | (65.2) |
| DISBEN3  | 5  | FDISBEN3 | DBER | A1 | 31  | (65.3) |
| DISBEN4  | 5  | FDISBEN4 | DBER | A1 | 21  | (66)   |
| INDP     | 2  | FINDPRIX | DBER | A1 | 10  | (67)   |
| IMPOS    | 7  | FIMPOS   | DBER | A1 | 53  | (68)   |

Relation IDET1 : 1.1

1 ET A 15 0 1 1 11 1  
 2 NOM A 44 0 1 12 40 0  
 3 ADR A 64 0 1 52 60 0  
 4 DATECR D1 14 0 7 112 10 0  
 5 EP A 14 0 1 122 11 0  
 6 SECTACT1 A 6 0 1 133 4 0  
 7 SECTACT2 A 6 0 1 137 4 0  
 8 SECTACT3 A 6 0 1 141 4 0

Relation IDET2 : 1.2

1 ET A 15 0 1 1 11 1  
 2 DATEDEX D1 12 0 7 12 10 0

Relation IDEP : 2

1 EP A 15 0 1 1 11 1  
 2 ETHR A 50 0 1 12 44 0  
 3 ADR A 64 0 1 56 60 0  
 4 NOM A 44 0 1 116 40 0  
 5 DATECR D1 12 0 7 156 10 0  
 6 PJUR A 5 0 1 166 3 0  
 7 SECTACT1 A 6 0 1 169 4 0  
 8 SECTACT2 A 6 0 1 173 4 0  
 9 SECTACT3 A 6 0 1 177 4 0

Relation DONSS : 3.1

1 EAT A 21 0 1 1 15 1  
 2 EA A 18 0 1 1 13 0  
 3 EP A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 MOISRO F 4 0 7 16 2 0  
 7 MONTANTO F 12 0 7 18 10 0

Relation DPP : 3.2

1 EAT A 21 0 1 1 15 1  
 2 EA A 18 0 1 1 13 0  
 3 EP A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 MOISRP F 4 0 7 16 2 0  
 7 MONTANTP F 12 0 7 18 10 0

Relation DIS : 3.3

1 EAT A 21 0 1 1 15 1  
 2 EA A 18 0 1 1 13 0  
 3 EP A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 MOISRI F 4 0 7 16 2 0  
 7 MONTANTI F 12 0 7 18 10 0

Relation LICENC : 4

1 EA A 19 0 1 1 13 1  
 2 EP A 15 0 1 1 11 0  
 3 AN A 4 0 1 12 2 0  
 4 NBLIC F 8 0 7 14 6 0

Relation CAPASSIF : 5

1 EA A 19 0 1 1 13 1  
 2 EP A 15 0 1 1 11 0  
 3 AN A 4 0 1 12 2 0  
 4 VALPAS F 12 0 7 14 10 0



Relation CAPP1 : 6.1

1 EA A 17 0 1 1 13 1  
 2 EP 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 FPERM F 12 0 7 14 10 0  
 5 FPRO F 12 0 7 24 10 0  
 6 CAPSOC F 12 0 7 34 10 0  
 7 CAPLIB F 12 0 7 44 10 0  
 8 PROV F 12 0 7 54 10 0  
 9 RES F 12 0 7 64 10 0  
 10 REPORTS F 12 0 7 74 10 0  
 11 BENET F 12 0 7 84 10 0

Relation CAPP2 : 6.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 SUBEQ F 12 0 7 14 10 0  
 5 ACTN F 8 0 7 24 6 0  
 6 ACTA F 8 0 7 30 6 0  
 7 APPACT F 5 0 7 36 4 0  
 8 RESPR F 5 0 7 40 4 0

Relation CAPP3 : 6.3

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 MOYACT F' 6 0 7 14 6 0

Relation CAPP4 : 6.4

1 EA A 18 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 EVB A 8 0 1 14 6 0

Relation CAPP5 : 7

1 ETA A 21 0 1 1 15 1  
 2 EC A 18 0 1 1 13 0  
 3 EP A 14 0 1 1 11 0  
 4 TYACT A 4 0 1 12 2 0  
 5 AN A 3 0 1 14 2 0  
 6 NBACT F 8 0 7 16 6 0  
 7 VALNOM F 8 0 7 22 6 0

Relation CAPP6 : 8

1 ENA A 25 0 1 1 19 1  
 2 EN A 22 0 1 1 17 0  
 3 EP A 14 0 1 1 11 0  
 4 ACTNOM A 8 0 1 12 6 0  
 5 AN A 3 0 1 18 2 0  
 6 NAT A 22 0 1 20 20 0  
 7 STATUT A 3 0 1 40 2 0  
 8 NBACT F 8 0 7 42 6 0

Relation CAPP7 : 9

1 EEA A 31 0 1 1 24 1  
 2 EE A 28 0 1 1 22 0  
 3 EP A 14 0 1 1 11 0  
 4 EPPART A 14 0 1 12 11 0  
 5 AN A 3 0 1 23 2 0  
 6 PCPART F 6 0 7 25 4 0

Relation CAPP8 : 10

1 EA A 17 0 1 1 13 1  
 2 EPSNI A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VAL F 12 0 7 14 10 0

Relation CAPE1 : 11.1

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VMLT F 12 0 7 14 10 0  
 5 VCT F 12 0 7 24 10 0

Relation CAPE2 : 11.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 4 0 1 12 2 0  
 3 VEC F 12 0 7 14 10 0  
 4 VENOF F 12 0 7 24 10 0  
 5 VEPC F 12 0 7 34 10 0  
 6 VELTMB F 12 0 7 44 10 0  
 7 VELTME F 12 0 7 54 10 0

Relation CAPE3 : 12

1 EEAMJ A 31 0 1 1 23 1  
 2 EEAM A 28 0 1 1 21 0  
 3 EEA A 25 0 1 1 19 0  
 4 EO A 22 0 1 1 17 0  
 5 EP A 14 0 1 1 11 0  
 6 EMPOBL A 8 0 1 12 6 0  
 7 AN A 3 0 1 18 2 0  
 8 MOIS A 3 0 1 20 2 0  
 9 JJ A 3 0 1 22 2 0  
 10 MONTANT F 12 0 7 24 10 0  
 11 TINT F 6 0 7 34 4 0  
 12 TERME D1 12 0 7 38 10 0  
 13 NBOBL F 8 0 7 48 6 0

Relation CAPE4 : 13

1 EEAMJ A 31 0 1 1 23 1  
 2 EEAM A 28 0 1 1 21 0  
 3 EEA A 25 0 1 1 19 0  
 4 EA A 22 0 1 1 17 0  
 5 EP A 14 0 1 1 11 0  
 6 EAMP A 8 0 1 12 6 0  
 7 AN A 3 0 1 18 2 0  
 8 MOIS A 3 0 1 20 2 0  
 9 JJ A 3 0 1 22 2 0  
 10 MONTANT F 12 0 7 24 10 0  
 11 TINT F 6 0 7 24 10 0  
 12 TERME D1 12 0 7 38 10 0  
 13 DEMISS D1 12 0 7 48 10 0  
 14 ORGAN A 22 0 1 58 20 0

Relation CAPACTIF : 14

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VALACT F 12 0 7 14 10 0

Relation CAPANIMM1 : 15.1

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 FCIRC F 12 0 7 14 10 0  
 5 FDISP F 12 0 7 24 10 0  
 6 FREAL F 12 0 7 34 10 0  
 7 TITRES F 12 0 7 44 10 0



Relation CAPANIM2 : 15.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 BONETAT F 12 0 7 14 10 0  
 5 PLPROV F 12 0 7 24 10 0

Relation CAPAIMM1 : 16

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 IMMOB F 12 0 7 14 10 0  
 5 ICORP F 12 0 7 24 10 0  
 6 PARTIC F 12 0 7 34 10 0  
 7 AUIIM F 12 0 7 44 10 0  
 8 AMIMM F 12 0 7 54 10 0  
 9 AMIC F 12 0 7 65 10 0

Relation CAPAIMM2 : 17

1 EEA A 31 0 1 1 24 1  
 2 EEP A 28 0 1 1 22 0  
 3 EP A 14 0 1 1 11 0  
 4 EEPART A 14 0 1 12 11 0  
 5 AN A 3 0 1 23 2 0  
 6 PCPART F 6 0 7 25 4 0

Relation CAPHYS1 : 18

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VALBAT F 12 0 7 14 10 0  
 5 NBBAT F 5 0 7 24 3 0  
 6 SURF F 12 0 7 27 10 0  
 7 AGE F 5 0 7 37 3 0  
 8 VMATR F 12 0 7 40 10 0  
 9 VEQOUT F 12 0 7 50 10 0  
 10 VSMOY F 12 0 7 60 10 0

Relation CAPHYS2 : 19

1 EMA A 21 0 1 1 15 1  
 2 EM A18 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 MATR A 4 0 1 12 2 0  
 5 AN A 3 0 1 14 2 0  
 6 NBMR F 8 0 7 16 6 0

Relation CAPHYS3 : 20

1 EUA A 25 0 1 1 19 1  
 2 EU A 22 0 1 1 17 0  
 3 ET A 14 0 1 1 11 0  
 4 UMR A 8 0 1 12 6 0  
 5 AN A 3 0 1 18 2 0  
 6 AGE F 4 0 7 20 3 0

Relation CAPHYS4 : 21

1 EQA A 21 0 1 1 15 1  
 2 EQ A 18 0 1 1 15 0  
 3 ET A 14 0 1 1 11 0  
 4 TEQ A 4 0 1 12 2 0  
 5 AN A 3 0 1 14 2 0  
 6 NBUN F 8 0 7 16 6 0

Relation CAPHYS5 : 22

1 EUA A 25 0 1 1 19 1  
 2 EVE A 22 0 1 1 17 0  
 3 ET A 14 0 1 1 11 0  
 4 UEQ A 8 0 1 12 6 0  
 5 AN A 3 0 1 18 2 0  
 6 AGE F 4 0 7 20 3 0

Relation RATFIN1 : 23.1

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 RENTCP F 6 0 7 14 4 0  
 5 SOLVMT F 6 0 7 18 4 0  
 6 INDFIN F 6 0 7 22 4 0  
 7 CAPREMB F 6 0 7 26 4 0  
 8 RAMIMM F 8 0 7 30 6 0  
 9 FROUL F 6 0 7 36 4 0  
 10 ROTCAP F 6 0 7 40 4 0  
 11 MARGEER F 12 0 7 44 10 0

Relation RATFIN2 : 23.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VAIMM F 12 0 7 14 10 0  
 5 TAFIN F 6 0 7 24 4 0

Relation RATFIN3 : 23.3

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 RCTA F 8 0 7 14 6 0

Relation RATFIN4 : 23.4

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 IEND F 8 0 7 14 6 0  
 5 ICAPBEN F 6 0 7 20 4 0  
 6 VANN F 12 0 7 24 10 0  
 7 PANNCFN F 6 0 7 34 4 0  
 8 RAMAN F 8 0 7 38 6 0

Relation CAPINV1 : 24.1

1 EA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VINV F 12 0 7 14 10 0  
 5 VPREV F 12 0 7 24 10 0

Relation CAPINV2 : 24.2

1 EA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 TAUX F 6 0 7 14 4 0

Relation CINVEST1 : 25.1

1 EIA A 20 0 1 1 15 1  
 2 EI A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 TYINV A 3 0 1 12 2 0  
 5 AN A 3 0 1 14 2 0  
 6 VAUGCAP F 12 0 7 16 10 0  
 7 VAUGBAN F 12 0 7 26 10 0  
 8 VAUGBAU F 12 0 7 36 10 0  
 9 VAUGBP F 12 0 7 46 10 0  
 10 RED F 12 0 7 56 10 0

Relation CINVEST2 : 25.2

1 EIA A 20 0 1 1 15 1  
 2 EI A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 TYINV A 3 0 1 12 2 0  
 5 AN A 3 0 1 14 2 0  
 6 VTINV F 12 0 7 16 10 0



Relation CINVEST3 : 25.3

1 EIA A 20 0 1 1 15 1  
 2 EI A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 TYINV A 3 0 1 12 2 0  
 5 AN A 3 0 1 14 2 0  
 6 VPREV F 12 0 7 16 10 0  
 7 DES A 44 0 1 26 40 0

Relation AINVA1 : 26.1

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VIINV F 12 0 7 14 10 0  
 5 NAT A 22 0 1 24 20 0  
 6 INTNOM F 6 0 7 44 4 0  
 7 PINT F 6 0 7 48 4 0  
 8 VSUBA F 12 0 7 52 10 0  
 9 VSUBB F 12 0 7 62 10 0

Relation AINVA2 : 26.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 EPAPP A 14 0 1 14 11 0  
 5 NAT A 22 0 1 25 20 0  
 6 VAL F 12 0 7 45 10 0

Relation AINVA3 : 26.3

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 DES A 42 0 1 14 40 0  
 5 VAL F 12 0 7 54 10 0  
 6 CETAT F 12 0 7 64 10 0  
 7 EPBEN F 12 0 7 74 10 0  
 8 CORGCR F 12 0 7 84 10 0  
 9 MONT F 12 0 7 94 10 0

Relation AINVA4 : 27.1

1 PA A 17 0 1 1 13 1  
 2 PME A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VSUBV F 12 0 7 14 10 0  
 5 VINV F 12 0 7 24 10 0

Relation AINVA5 : 27.2

1 PA A 17 0 1 1 13 1  
 2 PME A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 CREDIT F 12 0 7 14 10 0  
 5 CONTRETAT F 12 0 7 24 10 0  
 6 GAR F 12 0 7 34 10 0

Relation AINVA6 : 28

1 PA A 18 0 1 1 13 1  
 2 PME A 14 0 1 1 11 0  
 3 AIDE A 4 0 1 12 2 0  
 4 DINT D1 12 0 7 14 10 0

Relation AINVA7 : 29

1 EAA A 21 0 1 1 15 1  
 2 EAI A 17 0 1 1 13 0  
 3 EP A 14 0 1 1 11 0  
 4 AIDE A 4 0 1 12 2 0  
 5 AN A 3 0 1 14 2 0  
 6 EPPME A 4 0 1 16 3 0  
 7 NDOSS A 12 0 1 19 10 0  
 8 ETBEN A 14 0 1 29 11 0  
 9 BUT A 22 0 1 40 20 0  
 10 DMP D1 12 0 7 60 10 0  
 11 NMENT F 8 0 7 70 6 0  
 12 DACH D1 12 0 7 76 10 0  
 13 NEMPC F 8 0 7 86 6 0  
 14 IDEB D1 12 0 7 92 10 0  
 15 DCLO D1 12 0 7 102 10 0

Relation NAIDINV3 : 30.3

1 EAAM A 24 0 1 1 17 1  
 2 EAA A 21 0 1 1 15 0  
 3 EAI A 18 0 1 1 13 0  
 4 EP A 14 0 1 1 11 0  
 5 AIDE A 4 0 1 12 2 0  
 6 AN A 3 0 1 14 2 0  
 7 MOIS A 3 0 1 16 2 0  
 8 IDAPP A 22 0 1 18 20 0  
 9 NAT A 22 0 1 38 20 0  
 10 NBEMPL F 8 0 7 58 6 0

Relation NAIDINV1 : 30.1

1 EEAM A 24 0 1 1 17 1  
 2 EAA A 21 0 1 1 15 0  
 3 EAI A 18 0 1 1 13 0  
 4 EP A 14 0 1 1 11 0  
 5 AIDE A 4 0 1 12 2 0  
 6 AN A 3 0 1 14 2 0  
 7 MOIS A 3 0 1 16 2 0  
 8 ETDEM A 14 0 1 18 11 0  
 9 VAL F 12 0 7 29 10 0  
 10 NAT A 22 0 1 39 20 0  
 11 BUT A 22 0 1 59 20 0  
 12 NEMPL F 8 0 7 79 6 0  
 13 DACH D1 12 0 7 85 10 0  
 14 COMMUNE A 22 0 1 95 20 0  
 15 ZONING A 22 0 1 115 20 0  
 16 CNOM A 22 0 1 135 20 0  
 17 CACT A 22 0 1 155 20 0  
 18 CLOCAL A 22 0 1 175 20 0  
 19 CHAT A 22 0 1 195 20 0  
 20 CVAL F 12 0 7 215 10 0  
 21 CNEMP F 8 0 7 225 6 0  
 22 CJUR A 5 0 1 231 3 0  
 23 CVCAPINV F 12 0 7 234 10 0

Relation DESPROD1 : 31.1

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 CONSCAP F 12 0 7 14 10 0

Relation DESPROD2 : 31.2

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 PCTVA F 6 0 7 14 4 0  
 5 PCCVA F 6 0 7 18 4 0

Relation NAIDINV2 : 30.2

1 EAAM A 24 0 1 1 17 1  
 2 EAA A 21 0 1 1 15 0  
 3 EAI A 18 0 1 1 13 0  
 4 EP A 14 0 1 1 11 0  
 5 AIDE A 4 0 1 12 2 0  
 6 AN A 3 0 1 14 2 0  
 7 MOIS A 3 0 1 16 2 0  
 8 ZONE A 22 0 1 18 20 0

Relation DESPROD3 : 32

1 EPAM A 26 0 1 1 19 1  
 2 EPA A 23 0 1 1 17 0  
 3 EPR A 20 0 1 1 15 0  
 4 ET A 14 0 1 1 11 0  
 5 PROD A 6 0 1 12 4 0  
 6 AN A 3 0 1 14 2 0



7 MOIS A 3 0 1 18 2 0  
 8 DUREE A 8 0 1 20 6 0  
 9 CAPAC A 12 0 1 26 10 0  
 10 TRANSP A 22 0 1 36 20 0

Relation DESPROD4 : 33.1

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VIB F 12 0 7 14 10 0  
 5 VIN F 12 0 7 24 10 0

Relation DESPROD5 : 33.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 TAMQRT F 6 0 1 14 4 0

Relation DESPROD6 : 34

1 EATC A 35 0 1 1 33 1  
 2 ETC A 33 0 1 1 31 0  
 3 ET A 14 0 1 1 11 0  
 4 TECHNOL A 22 0 1 12 20 0  
 5 AN A 3 0 1 32 2 0  
 6 DES A 62 0 1 34 60 0

Relation CARPH1 : 35.1

1 BAT A 20 0 1 1 15 1  
 2 BA A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 NBTRAV F 8 0 7 16 6 0

7 NBMAN F 8 0 7 22 6 0  
 8 NBINT F 8 0 7 28 6 0  
 9 NBHOM F 8 0 7 34 6 0  
 10 NBFEM F 8 0 7 40 6 0  
 11 NBMVU F 8 0 7 46 6 0  
 12 NBEVUC F 8 0 7 52 6 0  
 13 NVPC F 8 0 7 58 6 0

Relation CARPH2 : 35.2

1 EAT A 18 0 1 1 15 1  
 2 EA A 15 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 TSynd F 6 0 7 16 4 0

Relation CARPH3 : 36

1 ESAT A 24 0 1 1 19 1  
 2 ESA A 22 0 1 1 17 0  
 3 ES A 19 0 1 1 15 0  
 4 ET A 14 0 1 1 11 0  
 5 SECTACT A 5 0 1 12 4 0  
 6 AN A 3 0 1 16 2 0  
 7 TRIM A 3 0 1 18 2 0  
 8 NBPNA F 8 0 7 20 6 0  
 9 NBTRAV F 8 0 7 26 6 0  
 10 NBMAN F 8 0 7 32 6 0  
 11 NBINT F 8 0 7 38 6 0

Relation CARPH4 : 37

1 EQAT A 25 0 1 1 18 1  
 2 EQA A 22 0 1 1 16 0  
 3 EQS A 19 0 1 1 14 0  
 4 ET A 14 0 1 1 11 0  
 5 QSCOL A 5 0 1 12 3 0  
 6 AN A 3 0 1 15 2 0  
 7 TRIM A 3 0 1 17 2 0  
 8 NBPERS F 8 0 7 19 6 0

Relation CARPH5 : 38.1

1 TRAV A 8 0 1 1 6 1  
 2 NOM A 42 0 1 7 40 0  
 3 ADR A 62 0 1 47 60 0

Relation CARPH6 : 38.2

1 TRAV A 8 0 1 1 6 1  
 2 ID A 20 0 1 7 20 0  
 3 DNAISS D1 12 0 1 27 10 0  
 4 SEXE A 2 0 1 37 1 0  
 5 NAT A 22 0 1 38 20 0  
 6 ETATC A 12 0 1 58 11 0

Relation CARPH7 : 38.3

1 TRAV A 8 0 1 1 6 1  
 2 QSCOL A 5 0 1 7 3 0  
 3 QEMPL A 5 0 1 10 3 0

Relation CARPH8 : 38.4

1 TRAV A 8 0 1 1 6 1  
 2 TTRAV A 1 70 1 7 15 0  
 3 QDEMP A 1 80 1 22 15 0

Relation CHARPH1 : 39.1

1 BAT A 20 0 1 1 15 1  
 2 BA A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 VCPTM F 12 0 7 16 10 0  
 7 VCPTI F 12 0 7 26 10 0  
 8 VCSTM F 12 0 7 36 10 0  
 9 VCSTI F 12 0 7 46 10 0

10 VRB F 12 0 7 56 10 0  
 11 VRBPO F 12 0 7 66 10 0  
 12 VRBPE F 12 0 7 76 10 0

Relation CHARPH2 : 39.2

1 BAT A 20 0 1 1 15 1  
 2 BA A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 MGD F 12 0 7 16 10 0  
 7 MADP F 12 0 7 26 10 0

Relation CHARPH3 : 39.3

1 BAT A 20 0 1 1 15 1  
 2 BA A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 TRIM A 3 0 1 14 2 0  
 6 VRPNA F 12 0 7 16 10 0  
 7 SYREM A 32 0 1 26 30 0

Relation CHARPH4 : 40

1 BQPAT A 25 0 1 1 18 1  
 2 EQA A 22 0 1 1 16 0  
 3 EQE A 19 0 1 1 14 0  
 4 ET A 14 0 1 1 11 0  
 5 QEMPL A 5 0 1 12 3 0  
 6 AN A 3 0 1 15 2 0  
 7 TRIM A 3 0 1 17 2 0  
 8 SALAB F 12 0 7 19 10 0  
 9 SUPPL F 12 0 7 29 10 0



Relation CHARPH5 : 41.1

1 TAT A 14 0 1 1 10 1  
 2 TA A 11 0 1 1 8 0  
 3 TRAV A 8 0 1 1 6 0  
 4 AN A 3 0 1 7 2 0  
 5 TRIM A 3 0 1 9 2 0  
 6 CONTRAT A 22 0 1 11 20 0  
 7 REMBA F 12 0 7 31 10 0  
 8 NBJR F 5 0 7 41 3 0  
 9 REGICJ A 5 0 1 44 3 0

Relation CHARPH6 : 41.2

1 TAT A 14 0 1 1 10 1  
 2 TA A 14 0 1 1 8 0  
 3 TRAV A 8 0 1 1 6 0  
 4 AN A 3 0 1 7 2 0  
 5 TRIM A 3 0 1 9 2 0  
 6 DUCONT A 4 0 1 11 3 0  
 7 HR F 5 0 7 14 4 0  
 8 HRN F 5 0 7 18 4 0  
 9 HRPF F 5 0 7 22 4 0  
 10 HRDE F 5 0 7 26 4 0  
 11 HRTE F 5 0 7 30 4 0  
 12 HRWE F 5 0 7 34 4 0  
 13 CH F 5 0 7 38 4 0

Relation AMELPH1 : 42

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 SPPER F 12 0 7 14 10 0  
 5 PREC F 8 0 7 24 6 0  
 6 CREC F 8 0 7 30 6 0  
 7 EREC F 8 0 7 36 6 0  
 8 OREC F 8 0 7 42 6 0

Relation AMELPH2 : 43

1 TA A 11 0 1 1 8 1  
 2 TRAV A 8 0 1 1 6 0  
 3 AN A 3 0 1 7 2 0  
 4 HFORM F 8 0 7 9 6 0  
 5 CHS F 8 0 7 15 6 0  
 6 FCH F 8 0 7 21 6 0  
 7 NOM A 22 0 1 27 20 0  
 8 DENTCF D1 12 0 7 47 10 0  
 9 QCOMP A 5 0 1 57 3 0  
 10 DOBTQ D1 12 0 7 60 10 0

Relation RENPH1 : 44

1 EOAMJ A 39 0 1 1 32 1  
 2 EEAM A 36 0 1 1 30 0  
 3 EEA A 33 0 1 1 28 0  
 4 EEO A 30 0 1 1 26 0  
 5 ET A 14 0 1 1 11 0  
 6 EMPOFF A 16 0 1 12 15 0  
 7 AN A 3 0 1 27 2 0  
 8 MOIS A 3 0 1 29 2 0  
 9 JJ A 3 0 1 31 2 0  
 10 DVAC D1 12 0 7 33 10 0  
 11 DES A 32 0 1 43 30 0  
 12 SOURCE A 22 0 1 73 20 0

Relation RENPH2 : 45.1

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 NBEO F 8 0 7 14 6 0

Relation RENPH3 : 45.2

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 NBEONO F 8 0 7 14 6 0  
 5 QEMPL A 5 0 1 20 3 0

Relation ACTPROD : 46

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VAPM F 12 0 7 14 10 0  
 5 VACF F 12 0 7 24 10 0  
 6 DEGITR A 12 0 1 34 10 0

Relation ACHCESS1 : 47.1

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VAPE F 12 0 7 14 10 0  
 5 VAPR F 12 0 7 24 10 0  
 6 VMPR F 12 0 7 34 10 0

Relation ACHCESS2 : 47.2

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VAMR F 12 0 7 14 10 0  
 5 VAMB F 12 0 7 24 10 0  
 6 VAME F 12 0 7 34 10 0  
 7 QAPE F 10 0 7 44 8 0  
 8 QAMR F 10 0 7 52 8 0  
 9 QAMB F 10 0 7 60 8 0  
 10 QAME F 10 0 7 68 8 0  
 11 REGION A 22 0 1 76 20 0  
 12 PAYS A 22 0 1 96 20 0

Relation STOCKS1 : 48.1

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VSDE F 12 0 7 14 10 0  
 5 VSPED F 12 0 7 24 10 0  
 6 VSPECDF 12 0 7 34 10 0  
 7 VSPSD F 12 0 7 44 10 0  
 8 VSPRD F 12 0 7 54 10 0  
 9 QSPSD F 10 0 7 64 8 0  
 10 VSFE F 12 0 7 72 10 0  
 11 VSPEF F 12 0 7 82 10 0  
 12 VSPECF F 12 0 7 92 10 0  
 13 VSPSF F 12 0 7 102 10 0  
 14 VSPRF F 12 0 7 112 10 0  
 15 QSPSF F 10 0 7 122 8 0

Relation STOCKS2 : 48.2

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 QSDE F 10 0 7 14 8 0  
 5 QSPED F 10 0 7 22 8 0  
 6 QSPECD F 10 0 7 30 8 0  
 7 QSFE F 10 0 7 38 8 0  
 8 QSPEF F 10 0 7 46 8 0  
 9 QSPECF F 10 0 7 54 8 0

Relation CONSOM1 : 49

1 EAM A 23 0 1 1 18 1  
 2 EMP A 20 0 1 1 16 0  
 3 ET A 14 0 1 1 11 0  
 4 MP A 6 0 1 12 5 0  
 5 AN A 3 0 1 17 2 0  
 6 QCONS F 10 0 7 19 8 0  
 7 VCONS F 10 0 7 27 8 0



Relation CONSON2 : 50

1 ESEA A 23 0 1 1 18 1  
 2 ESE A 20 0 1 1 16 0  
 3 ET A 14 0 1 1 11 0  
 4 SEN A 6 0 1 12 5 0  
 5 AN A 3 0 1 17 2 0  
 6 QCONS F 10 0 7 19 8 0  
 7 VCONS F 10 0 7 27 8 0

Relation CONSON3 : 51.1

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VGAC F 12 0 7 14 10 0  
 5 VCE F 12 0 7 24 10 0  
 6 VCER F 12 0 7 34 10 0  
 7 VCME F 12 0 7 44 10 0  
 8 VCPO F 12 0 7 54 10 0  
 9 QGAC F 10 0 7 64 8 0  
 10 QCE F 10 0 7 72 8 0  
 11 VET F 12 0 7 80 10 0  
 12 VTST F 12 0 7 90 10 0  
 13 VCHFD F 12 0 7 100 10 0  
 14 VDER F 12 0 7 110 10 0  
 15 VPAV F 12 0 7 120 10 0  
 16 VLBFP F 12 0 7 130 10 0  
 17 VLBEQ F 12 0 7 140 10 0  
 18 VFB F 12 0 7 150 10 0

Relation CONSON4 : 51.2

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 QCME F 10 0 7 14 8 0  
 5 SSST A 35 0 1 22 33 0

Relation PROD1 : 52.1

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 LPF A 52 0 1 14 50 0  
 5 LPS A 22 0 1 64 20 0

Relation PRGD2 : 52.2

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 INDEP F 5 0 7 14 4 0

Relation PRGD3 : 52.3

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VTLM F 12 0 7 14 10 0

Relation PRGD4 : 53.1

1 EPA A 23 0 1 1 17 1  
 2 EPR A 20 0 1 1 15 0  
 3 ET A 14 0 1 1 11 0  
 4 PROD A 5 0 1 12 4 0  
 5 AN A 3 0 1 16 2 0  
 6 QFLM F 10 0 7 18 8 0  
 7 QFT F 10 0 7 26 8 0  
 8 QFA F 10 0 7 34 8 0  
 9 QC F 10 0 7 42 8 0  
 10 QPFC F 10 0 7 50 8 0

Relation PROD5 : 53.2

1 EPA A 23 0 1 1 17 1  
 2 EPR A 20 0 1 1 15 0  
 3 ET A 14 0 1 1 11 0  
 4 PROD A 6 0 1 12 4 0  
 5 AN A 3 0 1 16 2 0  
 6 EPQP F 12 0 7 18 10 0  
 7 VQP F 12 0 7 28 10 0

Relation VENTESV1 54.1

1 EPA A 23 0 1 1 17 1  
 2 EPR A 20 0 1 1 15 0  
 3 ET A 14 0 1 1 11 0  
 4 PROD A 6 0 1 12 4 0  
 5 AN A 3 0 1 16 2 0  
 6 VVTC F 12 0 7 18 10 0  
 7 VVDE F 12 0 7 28 10 0  
 8 QVEX F 10 0 7 38 8 0  
 9 QVE F 10 0 7 46 8 0

Relation VENTESV2 : 54.2

1 EPA A 23 0 1 1 17 1  
 2 EPR A 20 0 1 1 15 0  
 3 ET A 14 0 1 1 11 0  
 4 PROD A 6 0 1 12 4 0  
 5 AN A 3 0 1 16 2 0  
 6 VVR F 12 0 7 18 10 0  
 7 VVAR F 12 0 7 28 10 0  
 8 QVR F 10 0 7 38 8 0  
 9 QVAR F 10 0 7 46 8 0  
 10 REGION A 22 0 1 54 20 0  
 11 PAYS A 22 0 1 74 20 0

Relation VENTESV3 : 55

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VVPST F 12 0 7 14 10 0  
 5 VPCA F 12 0 7 24 10 0  
 6 VTET F 12 0 7 34 10 0  
 7 VPA F 12 0 7 44 10 0

Relation POLL1 : 56

1 BAM A 21 0 1 1 15 1  
 2 BA A 17 0 1 1 13 0  
 3 ET A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 MOIS A 3 0 1 14 2 0  
 6 NBPLAI F 8 0 7 16 7 0  
 7 AMENDES F 12 0 7 23 10 0

Relation POLL2 : 57

1 EYAM A 26 0 1 1 19 1  
 2 EYA A 23 0 1 1 17 0  
 3 EY A 20 0 1 1 15 0  
 4 ET A 14 0 1 1 11 0  
 5 TYPOLL A 6 0 1 12 4 0  
 6 AN A 3 0 1 16 2 0  
 7 MOIS A 3 0 1 18 2 0  
 8 MESURE A 22 0 1 20 20 0

Relation POLL3 : 58

1 EAM A 21 0 1 1 15 1  
 2 EA A 17 0 1 1 13 0  
 3 EPILL A 14 0 1 1 11 0  
 4 AN A 3 0 1 12 2 0  
 5 MOIS A 3 0 1 14 2 0  
 6 PETAT F 7 0 7 16 5 0



Relation POLL4 : 59

1 TAM A 12 0 1 1 9 1  
 2 TA A 9 0 1 1 7 0  
 3 TYPOLL A 6 0 1 1 5 0  
 4 AN A 3 0 1 6 2 0  
 5 MOIS A 3 0 1 8 2 0  
 6 MAIDES F 12 0 7 10 10 0  
 7 LOCEPU A 32 0 1 20 30 0  
 8 RECH A 62 0 1 50 60 0

Relation STRPRG : 60

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 PREV F 12 0 7 14 10 0  
 5 PPRMP F 6 0 7 24 4 0  
 6 PPRMAC F 6 0 7 28 4 0  
 7 PPRSACH F 6 0 7 32 4 0  
 8 PPREN F 6 0 7 36 4 0  
 9 PPRAM F 6 0 7 40 4 0  
 10 PPRFG F 6 0 7 44 4 0  
 11 PPREV F 6 0 7 48 4 0  
 12 PPRDIV F 6 0 7 52 4 0

Relation STRPRPRO : 61

1 EPA A 23 0 1 1 17 1  
 2 EPR A 20 0 1 1 15 0  
 3 ET A 14 0 1 1 11 0  
 4 PROD A 6 0 1 12 4 0  
 5 AN A 3 0 1 16 2 0  
 6 PVEU F 12 0 7 18 10 0  
 7 PREV F 12 0 7 28 10 0  
 8 VMP F 12 0 7 38 10 0  
 9 VMAC F 12 0 7 48 10 0  
 10 VSACH F 12 0 7 58 10 0  
 11 VEN F 12 0 7 68 10 0  
 12 VAM F 12 0 7 78 10 0  
 13 VFG F 12 0 88 10 0  
 14 VFV F 12 0 7 98 10 0  
 15 VDIV F 12 0 7 108 10 0  
 16 MBEN F 12 0 7 118 10 0  
 17 DAPPL D1 12 0 7 128 10 0

Relation PRODUCTI : 62

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 PMVO F 12 0 7 14 10 0  
 5 PMVA F 12 0 7 24 10 0  
 6 DEPTR F 12 0 7 34 10 0  
 7 PRM F 12 0 7 44 10 0  
 8 IMPR F 5 0 7 54 4 0  
 9 VAMT F 12 0 7 58 10 0  
 10 IVAMT F 5 0 7 68 4 0

Relation RESEX1 : 63.1

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VCFN F 12 0 7 14 10 0  
 5 VPB F 12 0 7 24 10 0  
 6 VAF F 12 0 7 34 10 0  
 7 VCAF F 12 0 7 44 10 0

Relation RESEX2 : 63.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VCHA F 12 0 7 14 10 0  
 5 VREX F 12 0 7 24 10 0  
 6 VPP F 12 0 7 34 10 0  
 7 VFEX F 12 0 7 44 10 0

Relation RESEX3 : 63.3

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VPPIN F 12 0 7 14 10 0  
 5 VCHF F 12 0 7 24 10 0  
 6 VAM F 12 0 7 34 10 0  
 7 VREP F 12 0 7 44 10 0

Relation RESEX4 : 64.1

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VSUBV F 12 0 7 14 10 0

Relation RESEX5 : 64.2

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VEXCB F 12 0 7 14 10 0

Relation RESEX6 : 64.3

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VPRIMES F 12 0 7 14 10 0  
 5 MASS F 12 0 7 24 10 0

Relation DISBEN1 : 65.1

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VDIV F 12 0 7 14 10 0  
 5 VTANT F 12 0 7 24 10 0  
 6 VDONS F 12 0 7 34 10 0  
 7 VBEN F 12 0 7 44 10 0

Relation DISBEN2 : 65.2

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VMS F 12 0 7 14 10 0  
 5 IEMVS F 8 0 1 24 6 0

Relation DISBEN3 : 65.3

1 EA A 17 0 1 1 13 1  
 2 EP A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 VDCV F 12 0 7 14 10 0  
 5 IEVOL F 8 0 7 24 8 0

Relation DISBEN4 : 66

1 EA A 17 0 1 1 13 1  
 2 E A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 PCVA F 6 0 7 14 4 0  
 5 PPVA F 6 0 7 18 4 0

Relation INDP : 67

1 AM A 6 0 1 1 4 1  
 2 AN A 3 0 1 1 2 0  
 3 MOIS A 3 0 1 3 2 0  
 4 IPD F 8 0 7 5 6 0

Relation IMPOS : 68

1 BA A 17 0 1 1 13 1  
 2 ET A 14 0 1 1 11 0  
 3 AN A 3 0 1 12 2 0  
 4 MDACH F 12 0 7 14 10 0  
 5 VTAV F 12 0 7 24 10 0  
 6 VTVAE F 12 0 7 34 10 0  
 7 VAI F 12 0 7 44 10 0



CONTENU DES RELATIONS DE LA  
=====

BASE DE DONNEES DESCRIPTIVE  
=====

## 1) Relation COMDATA (DATAID, FORMULE, PROGR)

```

=====
EVB 424009 PC50
EPQP 213001213005 PC28
VACF 210002423007432012 PC46
DEGTR 210003213001 PC47
VAPM 213001213005213014213015213016215006 PC48
VAPM 214012214013214032214004211001211005 PC48
VAPM 214014214005212015214002214011212016 PC48
VAPM 212017212018212019212020212021212022 PC48
DEPTR 312005311001 PC30
PRM 313005313006 PC31
IMPR 313007 PC32
VAMT 210003311001 PC33
IVAMT 313009 PC34
VCFN 423002423003423004423012 PC35
VAF 424004423004 PC36
VCAF 423013424001 PC37
VPB 423007423008423009 PC38
VEXCB 210003312005 PC39
IEVOL 424005 PC40
VMS 312005111101 PC41
IEVMS 424007 PC42
VDCV 424001111101 PC43
PCVA 312005210003 PC44
PPVA 423007210003 PC45
VQP 313003215001215002 PC29
RENTCP 423007422002 PC7
SOLVMLT 422002422009 PC8
INDFIN 422002422016 PC9
CAPREMB 422010423014 PC10
RAMIMM 421005421001 PC11
FROUL 421001422001421005 PC12
ROTCAP 423001422002 PC13
MARGEBR 423007423001 PC14
VAIMM 210003210001 PC15
TAFIN 424004423004324007 PC16
RCTA 312005311001423001 PC17
IEND 423006210003 PC18
ICAPBEN 423010210003 PC19
PANNCFN 425013423007 PC20
RAMAN 421005425013 PC21
TAUX 324007210003 PC22
VTINV 324002324006 PC23
PCTVA 312005210003 PC24
PCCVA 323003210003 PC25
VIB 421001311001 PC26
VIN 421001421005311001 PC27
PMVO 215001215002311001 PC51
PMVA 311001313004 PC52
INDPM 313001 PC53

```



## 2) Relation COMPOSANT (RELPHID, DATAID, NUMDATA)

```

=====
IDET1 NOM 110001
IDET1 ADR 110002
IDET1 DATECR 110003
IDET1 EP 110004
IDET1 SECTACT1 120101
IDET1 SECTACT2 120102
IDET1 SECTACT3 120003
IDET2 DATEDEX 210001
IDEP ETHR 110005
IDEP ADR 110006
IDEP NOM 110007
IDEP DATECR 110008
IDEP FJUR 110009
IDEP SECTACT1 120001
IDEP SECTACT2 120002
IDEP SECTACT3 120003
DONSS MONTANTO 440002
DONSS MOISRO 440001
DPP MOISRP 440003
DPP MONTANTP 440004
DIS MOISRI 440005
DIS MONTANTI 440006
LICENC NBLC 440007
CAPASSIF VALPAS 422017
CAPP1 FPERM 422001
CAPP1 FPRO 422002
CAPP1 CAPSOC 422003
CAPP1 CAPLIB 422004
CAPP1 RES 422005
CAPP1 PROV 422006
CAPP1 REPORTS 422007
CAPP1 BENET 422008
CAPP2 SUBEQ 422009
CAPP2 ACTN 422020
CAPP2 ACTA 422024
CAPP2 APPACT 422028
CAPP2 RESPR 422029
CAPP3 MOYACT 424009
CAPP4 EVB 424010
CAPP5 NBACT 422018
CAPP5 VALNOM 422019
CAPP6 NAT 422021
CAPP6 STATUT 422022
CAPP6 NBACT 422023
CAPP7 PCPART 422031
CAPP8 VAL 414029
CAPE1 VMLT 422010
CAPE1 VCT 422016
CAPE2 VEO 422011
CAPE2 VENOP 422012
CAPE2 VEPC 422013
CAPE2 VELTMB 422014
CAPE2 VELTME 422015
CAPE3 MONTANT 422032

```

|           |         |        |
|-----------|---------|--------|
| CAPE3     | TINT    | 422033 |
| CAPE3     | TERME   | 422034 |
| CAPE3     | NBOBL   | 422036 |
| CAPE4     | MONTANT | 422037 |
| CAPE4     | TINT    | 422038 |
| CAPE4     | TERME   | 422039 |
| CAPE4     | ORGAN   | 422041 |
| CAPACTIF  | VALACT  | 421010 |
| CAPANIMM1 | FCIRC   | 421007 |
| CAPANIMM1 | FDISP   | 421008 |
| CAPANIMM1 | FREAL   | 421009 |
| CAPANIMM1 | TITRES  | 421001 |
| CAPANIMM2 | BO?ETAT | 421012 |
| CAPANIMM2 | PLPROV  | 421013 |
| CAPAIMM1  | IMMOB   | 421001 |
| CAPAIMM1  | ICORP   | 421002 |
| CAPAIMM1  | PARTIC  | 421003 |
| CAPAIMM1  | AUIMM   | 421004 |
| CAPAIMM1  | AMIMM   | 421005 |
| CAPAIMM1  | AMIC    | 421006 |
| CAPAIMM2  | PCPART  | 421015 |
| CAPHYS1   | VALBAT  | 321001 |
| CAPHYS1   | NBBAT   | 321002 |
| CAPHYS1   | SURF    | 321003 |
| CAPHYS1   | AGE     | 321004 |
| CAPHYS1   | VMATR   | 321005 |
| CAPHYS1   | VEQOUT  | 321008 |
| CAPHYS1   | VSMOY   | 321001 |
| CAPHYS2   | NBMR    | 321006 |
| CAPHYS3   | AGE     | 321007 |
| CAPHYS4   | NBUN    | 321009 |
| CAPHYS5   | AGE     | 321010 |
| RATFIN1   | RENTCP  | 425001 |
| RATFIN1   | SOLVMLT | 425003 |
| RATFIN1   | INDFIN  | 425004 |
| RATFIN1   | CAPREMB | 425006 |
| RATFIN1   | RAMIMM  | 425009 |
| RATFIN1   | FROUL   | 425010 |
| RATFIN1   | ROTCAP  | 425011 |
| RATFIN1   | MARGEBR | 425012 |
| RATFIN2   | VAIMM   | 425002 |
| RATFIN2   | TAFIN   | 425008 |
| RATFIN3   | RCTA    | 425016 |
| RATFIN4   | IEND    | 425005 |
| RATFIN4   | ICAPBEN | 425007 |
| RATFIN4   | VANN    | 425013 |
| RATFIN4   | PANNCFN | 425014 |
| RATFIN4   | RAMAN   | 425015 |
| CAPINV1   | VINV    | 324001 |
| CAPINV1   | VPREV   | 324009 |
| CAPINV2   | TAUX    | 324008 |
| CINVEST1  | VAUGCAP | 324002 |
| CINVEST1  | VAUGBAN | 324003 |
| CINVEST1  | VAUGBAU | 324004 |
| CINVEST1  | VAUGBP  | 324005 |
| CINVEST1  | RED     | 324006 |
| CINVEST2  | VTINV   | 324007 |



|          |           |        |
|----------|-----------|--------|
| CINVEST3 | VPREV     | 324010 |
| CINVEST3 | DES       | 324011 |
| AINVA1   | VIINV     | 414005 |
| AINVA1   | NAT       | 414006 |
| AINVA1   | INTNOM    | 414016 |
| AINVA1   | PINT      | 414017 |
| AINVA1   | VSUBA     | 414018 |
| AINVA1   | VSUBB     | 414019 |
| AINVA2   | EPAPP     | 414025 |
| AINVA2   | NAT       | 414026 |
| AINVA2   | VAL       | 414027 |
| AINVA3   | DES       | 414007 |
| AINVA3   | VAL       | 414020 |
| AINVA3   | CETAT     | 414021 |
| AINVA3   | EPBEN     | 414022 |
| AINVA3   | CORGCR    | 414023 |
| AINVA3   | MONT      | 414024 |
| AINVA4   | VSUBV     | 414203 |
| AINVA4   | VINV      | 414204 |
| AINVA5   | CREDIT    | 414205 |
| AINVA5   | CONTRETAT | 414206 |
| AINVA5   | GAR       | 414207 |
| AINVA6   | DINT      | 414209 |
| AINVA7   | EPPME     | 414003 |
| AINVA7   | NDOSS     | 414004 |
| AINVA7   | ETBEN     | 414008 |
| AINVA7   | BUT       | 414009 |
| AINVA7   | DMP       | 414010 |
| AINVA7   | NMENT     | 414011 |
| AINVA7   | DACH      | 414012 |
| AINVA7   | NEMPC     | 414013 |
| AINVA7   | DDEB      | 414014 |
| AINVA7   | DCLO      | 414015 |
| NAIDINV1 | VAL       | 414104 |
| NAIDINV1 | ETDEM     | 414102 |
| NAIDINV1 | NAT       | 414105 |
| NAIDINV1 | BUT       | 414106 |
| NAIDINV1 | NEMPL     | 414107 |
| NAIDINV1 | DACH      | 414108 |
| NAIDINV1 | COMMUNE   | 414109 |
| NAIDINV1 | ZONING    | 414110 |
| NAIDINV1 | CNOM      | 414118 |
| NAIDINV1 | CACT      | 414119 |
| NAIDINV1 | CLOCAL    | 414120 |
| NAIDINV1 | CNAT      | 414121 |
| NAIDINV1 | CVAL      | 414122 |
| NAIDINV1 | CNEMP     | 414123 |
| NAIDINV1 | CJFUR     | 414124 |
| NAIDINV1 | CVCAPINV  | 414125 |
| NAIDINV2 | ZONE      | 414111 |
| NAIDINV3 | IDAPP     | 414112 |
| NAIDINV3 | NAT       | 414113 |
| NAIDINV3 | NBEMPL    | 414114 |
| DESPROD1 | CONSCAP   | 323003 |
| DESPROD2 | PCTVA     | 323004 |
| DESPROD2 | PCCVA     | 323005 |
| DESPROD3 | DUREE     | 322004 |

|          |         |        |
|----------|---------|--------|
| DESPROD3 | CAPAC   | 322005 |
| DESPROD3 | TRANSP  | 322006 |
| DESPROD3 | DEBUT   | 222003 |
| DESPROD4 | VIB     | 323001 |
| DESPROD4 | VIN     | 323002 |
| DESPROD5 | TAMORT  | 323006 |
| DESPROD6 | DES     | 322003 |
| DESPROD6 | DDES    | 322002 |
| CARPH1   | NBTRAV  | 311001 |
| CARPH1   | NBMAN   | 311002 |
| CARPH1   | NBINT   | 311003 |
| CARPH1   | NBHOM   | 311005 |
| CARPH1   | NBFEM   | 311006 |
| CARPH1   | NBMVU   | 311007 |
| CARPH1   | NBEVUC  | 311008 |
| CARPH1   | NVPC    | 311009 |
| CARPH2   | TSYND   | 311014 |
| CARPH3   | NBPNA   | 311004 |
| CARPH3   | NBTRAV  | 311011 |
| CARPH3   | NBMAN   | 311012 |
| CARPH3   | NBINT   | 311013 |
| CARPH4   | NBPERS  | 311010 |
| CARPH5   | NOM     | 311101 |
| CARPH5   | ADR     | 311107 |
| CARPH6   | ID      | 311102 |
| CARPH6   | DNAISS  | 311103 |
| CARPH6   | SEXE    | 311104 |
| CARPH6   | NAT     | 311105 |
| CARPH6   | ETATC   | 311106 |
| CARPH7   | QSCOL   | 311109 |
| CARPH7   | QEMPL   | 311110 |
| CARPH8   | TTRAV   | 311108 |
| CARPH8   | QDEMP   | 311111 |
| CHARPH1  | VCPTM   | 312001 |
| CHARPH1  | VCPTI   | 312002 |
| CHARPH1  | VCSTM   | 312003 |
| CHARPH1  | VCSTI   | 312004 |
| CHARPH1  | VRB     | 312006 |
| CHARPH1  | VRBPO   | 312007 |
| CHARPH1  | VRBPE   | 312008 |
| CHARPH2  | MGDP    | 312005 |
| CHARPH2  | MADP    | 312013 |
| CHARPH3  | VRPNA   | 312009 |
| CHARPH3  | SYREM   | 312012 |
| CHARPH4  | SALAB   | 312010 |
| CHARPH4  | SUPPL   | 312011 |
| CHARPH5  | CONTRAT | 312101 |
| CHARPH5  | REMB    | 312104 |
| CHARPH5  | NBJR    | 312105 |
| CHARPH5  | REGICJ  | 312113 |
| CHARPH6  | DCCONT  | 312102 |
| CHARPH6  | DUCONT  | 312103 |
| CHARPH6  | HR      | 312106 |
| CHARPH6  | HRW     | 312107 |
| CHARPH6  | HRPE    | 312108 |
| CHARPH6  | HRDE    | 312109 |
| CHARPH6  | HRTE    | 312110 |
| CHARPH6  | HRWE    | 312111 |



|          |        |        |
|----------|--------|--------|
| CHARPH6  | CH     | 312112 |
| AMELPH1  | SFPER  | 315001 |
| AMELPH1  | PREC   | 315002 |
| AMELPH1  | CREC   | 315003 |
| AMELPH1  | EREC   | 315004 |
| AMELPH1  | OREC   | 315005 |
| AMELPH2  | HFORM  | 315101 |
| AMELPH2  | CMS    | 315102 |
| AMELPH2  | FCH    | 315103 |
| AMELPH2  | NOMCF  | 315104 |
| AMELPH2  | DENTCF | 315105 |
| AMELPH2  | QCOMP  | 315106 |
| RENPH1   | DES    | 311203 |
| RENPH1   | SOURCE | 311204 |
| RENPH2   | NBEO   | 311201 |
| RENPH3   | NBEONO | 311205 |
| RENPH3   | QEMPL  | 311206 |
| ACTPROD  | VAPM   | 210002 |
| ACTPROD  | VACF   | 210003 |
| ACTPROD  | DEGTR  | 210004 |
| ACHCESS1 | VAPE   | 211001 |
| ACHCESS1 | VAPR   | 211005 |
| ACHCESS1 | VMPR   | 211012 |
| ACHCESS2 | VAMR   | 211002 |
| ACHCESS2 | VAMB   | 211003 |
| ACHCESS2 | VAME   | 211004 |
| ACHCESS2 | QAPE   | 211006 |
| ACHCESS2 | QAMR   | 211007 |
| ACHCESS2 | QAMB   | 211008 |
| ACHCESS2 | QAME   | 211009 |
| ACHCESS2 | REGION | 211010 |
| ACHCESS2 | PAYS   | 211011 |
| STOCKS1  | VSDE   | 214001 |
| STOCKS1  | VSPED  | 214002 |
| STOCKS   | VSPECD | 214003 |
| STOCKS1  | VSPSD  | 214004 |
| STOCKS1  | VSPRD  | 214005 |
| STOCKS1  | QSPSD  | 214009 |
| STOCKS1  | VCFE   | 214010 |
| STOCKS1  | VSPEF  | 214011 |
| STOCKS1  | VSPECF | 214012 |
| STOCKS1  | VSPSF  | 214013 |
| STOCKS1  | VSPRF  | 214014 |
| STOCKS1  | QSPSF  | 214018 |
| STOCKS2  | QSDE   | 214006 |
| STOCKS2  | QSPED  | 214007 |
| STOCKS2  | QSPECD | 214008 |
| STOCKS2  | QSFE   | 214015 |
| STOCKS2  | QSPEF  | 214016 |
| STOCKS2  | QSPECF | 214017 |
| CONSOM1  | QCONS  | 212002 |
| CONSOM1  | VCONS  | 212003 |
| CONSOM2  | QCONS  | 212005 |
| CONSOM2  | VCONS  | 212006 |
| CONSOM3  | VGAC   | 212007 |
| CONSOM3  | VCE    | 212008 |
| CONSOM3  | VCER   | 212009 |
| CONSOM3  | VCMER  | 212010 |

|          |         |        |
|----------|---------|--------|
| STRPRPRO | VFG     | 221008 |
| STRPRPRO | VFV     | 221009 |
| STRPRPRO | VDIV    | 221010 |
| STRPRPRO | MBEN    | 221011 |
| STRPRPRO | DAPPL   | 221012 |
| PRODUCTI | PMVO    | 313001 |
| PRODUCTI | PMVA    | 313005 |
| PRODUCTI | DEPTR   | 313006 |
| PRODUCTI | PRM     | 313007 |
| PRODUCTI | IMPR    | 313008 |
| PRODUCTI | VAMT    | 313009 |
| PRODUCTI | IVAMT   | 313010 |
| RESEX1   | VCFN    | 423011 |
| RESEX1   | VAF     | 423013 |
| RESEX1   | VACF    | 423014 |
| RESEX1   | VPB     | 423010 |
| RESEX2   | VCHA    | 423001 |
| RESEX2   | VREX    | 423002 |
| RESEX2   | VPP     | 423004 |
| RESEX2   | VFEX    | 423005 |
| RESEX3   | VAM     | 423003 |
| RESEX3   | VREP    | 423012 |
| RESEX3   | VPFIN   | 423008 |
| RESEX3   | VCHF    | 423009 |
| RESEX4   | VSUBV   | 423006 |
| RESEX5   | VEXCB   | 423007 |
| RESEX6   | VPRIMES | 324012 |
| RESEX6   | MASS    | 324013 |
| DISBEN1  | VDIV    | 424001 |
| DISBEN1  | VTANT   | 424002 |
| DISBEN1  | VDONS   | 424003 |
| DISBEN1  | VBEN    | 424004 |
| DISBEN2  | VMS     | 424007 |
| DISBEN2  | IEVMS   | 424008 |
| DISBEN3  | IEVOL   | 424006 |
| DISBEN3  | VDCV    | 424005 |
| DISBEN4  | PCVA    | 424011 |
| DISBEN4  | PPVA    | 424012 |
| INDP     | IPD     | 111101 |
| VENTESV3 | VPCA    | 213014 |
| VENTESV3 | VTET    | 213015 |
| VENTESV3 | VPA     | 213016 |
| IMPOS    | MDACH   | 432009 |
| IMPOS    | VTVAV   | 432010 |
| VMPOS    | VTVAE   | 432011 |
| IMPOS    | VAII    | 432012 |

### 3) Relation DATATYPE (DATAID,PRIMAIRE)

=====

|          |   |
|----------|---|
| ROTCAP   | C |
| VCHA     | P |
| MARGE BR | C |
| VAIMM    | C |
| VACF     | D |
| TAFIN    | C |



|          |         |        |
|----------|---------|--------|
| CONSUM3  | VCPO    | 212011 |
| CONSUM3  | QGAC    | 212012 |
| CONSUM3  | QCE     | 212013 |
| CONSUM3  | VFT     | 212015 |
| CONSUM3  | VST     | 212016 |
| CONSUM3  | VCHFD   | 212017 |
| CONSUM3  | VDER    | 212018 |
| CONSUM3  | VPAV    | 212019 |
| CONSUM3  | VLBFP   | 212020 |
| CONSUM3  | VLBEQ   | 212021 |
| CONSUM3  | VFB     | 212022 |
| CONSUM4  | VCMER   | 212014 |
| CONSUM4  | SSTR    | 212023 |
| PROD1    | LPF     | 222001 |
| PROD1    | LPS     | 222002 |
| PROD2    | INDPM   | 313002 |
| PROD3    | VTLM    | 215006 |
| PROD4    | QFLM    | 215001 |
| PROD4    | QFT     | 215002 |
| PROD4    | QFA     | 215003 |
| PROD4    | QC      | 215004 |
| PROD4    | QPEC    | 215005 |
| PROD5    | EPQP    | 313003 |
| PROD5    | VQP     | 313004 |
| VENTESV1 | VVTC    | 213001 |
| VENTESV1 | VVDE    | 213004 |
| VENTESV1 | QVEX    | 213006 |
| VENTESV1 | QVE     | 213009 |
| VENTESV2 | VVR     | 213002 |
| VENTESV2 | VVAR    | 213003 |
| VENTESV2 | QVR     | 213007 |
| VENTESV2 | QVAR    | 213008 |
| VENTESV2 | REGION  | 213010 |
| VENTESV2 | PAYS    | 213011 |
| VENTESV3 | VVPST   | 213005 |
| POLL1    | NBPLAI  | 240003 |
| POLL1    | AMENDES | 240004 |
| POLL2    | MESURE  | 240002 |
| POLL3    | PETAT   | 240007 |
| POLL4    | MAIDES  | 240005 |
| POLL4    | LOCEPU  | 240006 |
| POLL4    | RECH    | 240008 |
| STRPRG   | PREV    | 221101 |
| STRPRG   | PPRMP   | 221102 |
| STRPRG   | PPRMAC  | 221103 |
| STRPRG   | PPRSACH | 221104 |
| STRPRG   | PPREN   | 221105 |
| STRPRG   | PPRAM   | 221106 |
| STRPRG   | PPRFG   | 221107 |
| STRPRG   | PPREV   | 221108 |
| STRPRG   | PPDDIV  | 221109 |
| STRPRPRO | PVEU    | 221001 |
| STRPRPRO | PREV    | 221002 |
| STRPRPRO | VMP     | 221003 |
| STRPRPRO | VMAC    | 221004 |
| STRPRPRO | VSACH   | 221005 |
| STRPRPRO | VEN     | 221006 |
| STRPRPRO | VAM     | 221007 |

|         |   |
|---------|---|
| VBEN    | P |
| VPP     | P |
| VTINV   | D |
| RCTA    | C |
| MGDP    | P |
| NBTRAV  | P |
| IEND    | C |
| VSUBV   | P |
| ICAPBEN | C |
| VPB     | D |
| VANN    | C |
| PANNCFN | C |
| RAMAN   | C |
| TAUX    | C |
| VAUGCAP | P |
| RED     | P |
| PCTVA   | C |
| PCCVA   | C |
| CONSCAP | P |
| VIB     | C |
| VIN     | C |
| EPQP    | D |
| VVPST   | P |
| VQP     | D |
| QFLM    | P |
| QFT     | P |
| DEPTR   | D |
| PRM     | D |
| IMPR    | C |
| VAMT    | D |
| IVAMT   | C |
| VCFN    | C |
| VREX    | P |
| VAM     | P |
| VREP    | P |
| VAF     | D |
| VDIV    | P |
| VPFIN   | P |
| VCHF    | P |
| VDCV    | D |
| VVTC    | P |
| IPD     | P |
| IEMVS   | C |
| VMS     | D |
| PSVA    | C |
| PPVA    | C |
| PMVA    | D |
| PMVO    | P |
| INDPM   | C |
| IEVOL   | C |
| VAII    | P |
| EVB     | C |
| MOYACT  | P |
| VPCA    | P |
| VTET    | P |
| VPA     | P |
| VTLM    | P |



|         |   |
|---------|---|
| VSPECF  | P |
| VSPSF   | P |
| VSPECD  | P |
| VSPSD   | P |
| VAPE    | P |
| VAPR    | P |
| VSPRF   | P |
| VSPRO   | P |
| VSPEF   | P |
| VSPED   | P |
| VFT     | P |
| VTST    | P |
| VCHFD   | P |
| VDER    | P |
| VPAV    | P |
| VLBFP   | P |
| VLBEQ   | P |
| VFB     | P |
| DEGTR   | C |
| VAPM    | C |
| FPERM   | P |
| VCT     | P |
| FPRO    | P |
| VMLT    | P |
| IMMOB   | P |
| AMIMM   | P |
| RENTCP  | C |
| VEXCB   | D |
| SOLVMLT | C |
| SUBEQ   | P |
| INDFIN  | C |
| CAPREMB | C |
| VCAF    | D |
| RAMIMM  | C |
| FROUL   | C |

#### 4) Relation KEYUNIT (KEYID,CARDINAL)

=====

|       |    |
|-------|----|
| EMP   | 10 |
| ESE   | 10 |
| ET    | 6  |
| EP    | 5  |
| TRAV  | 8  |
| EC    | 10 |
| EN    | 10 |
| EE    | 10 |
| EPSNI | 10 |
| EO    | 10 |
| EA    | 10 |
| EEP   | 10 |
| EM    | 10 |
| EU    | 10 |
| EQ    | 10 |
| EUE   | 10 |

|        |    |
|--------|----|
| EI     | 10 |
| PME    | 10 |
| PA     | 10 |
| EAI    | 10 |
| ETE    | 10 |
| EPR    | 10 |
| ES     | 10 |
| EQS    | 10 |
| EE0    | 10 |
| EY     | 10 |
| EPILL  | 10 |
| TYPOLL | 4  |
| EQE    | 10 |
| AM     | 1  |

5) Relation RELLOG (RELOGID,MVT,NBMXVR,KEYID)

|          |     |    |       |
|----------|-----|----|-------|
| IDET     | EX  |    | ET    |
| IDEP     | EX  |    | EP    |
| DETTES   | SIT | 12 | EP    |
| LICENC   | SIT | 3  | EP    |
| CAPASSIF | SIT | 3  | EP    |
| CAPP     | SIT | 3  | EP    |
| CAPP5    | SIT | 3  | EC    |
| CAPP6    | SIT | 3  | EN    |
| CAPP7    | SIT | 3  | EE    |
| CAPP8    | SIT | 3  | EPSNI |
| CAPE     | SIT | 3  | EP    |
| CAPE3    | EV  | 10 | EO    |
| CAPE4    | EV  | 10 | EA    |
| CAPACTIF | SIT | 3  | EP    |
| CAPANIMM | SIT | 3  | EP    |
| CAPAIMM1 | SIT | 3  | EP    |
| CAPAIMM2 | SIT | 3  | EEP   |
| CAPHYS1  | SIT | 3  | ET    |
| CAPHYS2  | SIT | 3  | EM    |
| CAPHYS3  | SIT | 3  | EU    |
| CAPHYS4  | SIT | 3  | EQ    |
| CAPHYS5  | SIT | 3  | EUE   |
| RATFIN   | SIT | 3  | EP    |
| CAPINV   | SIT | 3  | ET    |
| CINVEST  | SIT | 3  | EI    |
| AINVAA   | SIT | 3  | EP    |
| AINVAB   | SIT | 3  | PME   |
| AINVA6   | EV  | 10 | PA    |
| AINVA7   | SIT | 3  | EAI   |
| NAIDINV  | EV  | 10 | EAI   |
| DESPRODA | SIT | 3  | ET    |
| DESPROD3 | EV  | 10 | EPR   |
| DESPRODB | SIT | 3  | EP    |
| DESPROD6 | EV  | 10 | ETC   |
| CARPHA   | SIT | 12 | ET    |
| CARPH3   | SIT | 12 | ES    |
| CARPH4   | SIT | 12 | EQS   |
| CARPHB   | EX  |    | TRAV  |



|          |     |    |        |
|----------|-----|----|--------|
| CHARPHA  | SIT | 12 | ET     |
| CHARPH4  | SIT | 12 | EQE    |
| CHARPHB  | SIT | 12 | TRAV   |
| AMELPH1  | SIT | 3  | ET     |
| AMELPH2  | EV  | 10 | TRAV   |
| RENPH1   | EV  | 10 | EE0    |
| RENPH    | SIT | 3  | ET     |
| ACTPROD  | SIT | 3  | ET     |
| ACHCESS  | SIT | 3  | ET     |
| STOCKS   | SIT | 3  | ET     |
| CONSUM1  | SIT | 3  | EMP    |
| CONSUM2  | SIT | 3  | ESE    |
| CONSUM   | SIT | 3  | ET     |
| PRODUA   | SIT | 3  | ET     |
| PRODUB   | SIT | 3  | EPR    |
| VENTESV  | SIT | 3  | EPR    |
| VENTESV3 | SIT | 3  | ET     |
| POLL1    | EV  | 10 | ET     |
| POLL2    | EV  | 10 | EY     |
| POLL3    | EV  | 10 | EPILL  |
| POLL4    | EV  | 10 | TYPOLL |
| STRPRG   | SIT | 3  | ET     |
| STRPRPRO | SIT | 3  | EPR    |
| PRODUCTI | SIT | 3  | ET     |
| RESEXA   | SIT | 3  | EP     |
| RESEXB   | SIT | 3  | ET     |
| DISBEN   | SIT | 3  | EP     |
| DISBEN4  | SIT | 3  | ET     |
| INDP     | SIT | 36 | AM     |
| IMPOS    | SIT | 3  | ET     |

6) Relation RELPHY (RELPHID,USERID,WRPASS,RELOGID,CALCULEE)

|           |    |     |          |   |
|-----------|----|-----|----------|---|
| IDET1     | U1 | EU1 | IDET     | 0 |
| IDET2     | U1 | EU1 | IDET     | 0 |
| IDEP      | U1 | EU1 | IDEP     | 0 |
| DONSS     | U1 | EU1 | DETTES   | 0 |
| DPP       | U1 | EU1 | DETTES   | 0 |
| DIS       | U1 | EU1 | DETTES   | 0 |
| LICENC    | U1 | EU1 | LICENS   | 0 |
| CAPASSIF  | U1 | EU1 | CAPASSIF | 0 |
| CAPP1     | U1 | EU1 | CAPP     | 0 |
| CAPP2     | U1 | EU1 | CAPP     | 0 |
| CAPP3     | U1 | EU1 | CAPP     | 1 |
| CAPP4     | U1 | EU1 | CAPP     | 0 |
| CAPP5     | U1 | EU1 | CAPP5    | 0 |
| CAPP6     | U1 | EU1 | CAPP6    | 0 |
| CAPP7     | U1 | EU1 | CAPP7    | 0 |
| CAPP8     | U1 | EU1 | CAPP8    | 0 |
| CAPE1     | U2 | EU2 | CAPE     | 0 |
| CAPE2     | U2 | EU2 | CAPE     | 0 |
| CAPE3     | U2 | EU2 | CAPE3    | 0 |
| CAPE4     | U2 | EU2 | CAPE4    | 0 |
| CAPACTIF  | U1 | EU1 | CAPACTIF | 0 |
| CAPANIMM1 | U1 | EU1 | CAPANIMM | 0 |

|           |    |     |          |   |
|-----------|----|-----|----------|---|
| CAPANIMM2 | U1 | EU1 | CAPANIMM | 0 |
| CAPAIMM1  | U1 | EU1 | CAPAIMM1 | 0 |
| CAPAIMM2  | U1 | EU1 | CAPAIMM2 | 0 |
| CAPHYS1   | U2 | EU2 | CAPHYS1  | 0 |
| CAPHYS2   | U2 | EU2 | CAPHYS2  | 0 |
| CAPHYS3   | U2 | EU2 | CAPHYS3  | 0 |
| CAPHYS4   | U2 | EU2 | CAPHYS4  | 0 |
| CAPHYS5   | U2 | EU2 | CAPHYS5  | 0 |
| RATFIN1   | U1 | EU1 | RATFIN   | 1 |
| RATFIN2   | U1 | EU1 | RATFIN   | 1 |
| RATFIN3   | U1 | EU1 | RATFIN   | 1 |
| RATFIN4   | U1 | EU1 | RATFIN   | 1 |
| CAPINV1   | U1 | EU1 | CAPINV   | 0 |
| CAPINV2   | U1 | EU1 | CAPINV   | 1 |
| CINVEST1  | U1 | EU1 | CINVEST  | 0 |
| CINVEST2  | U1 | EU1 | CINVEST  | 1 |
| CINVEST3  | U1 | EU1 | CINVEST  | 0 |
| AINVA1    | U1 | EU1 | AINVAA   | 0 |
| AINVA2    | U1 | EU1 | AINVAA   | 0 |
| AINVA3    | U1 | EU1 | AINVAA   | 0 |
| AINVA4    | U1 | EU1 | AINVAB   | 0 |
| AINVA5    | U1 | EU1 | AINVAB   | 0 |
| AINVA6    | U1 | EU1 | AINVA6   | 0 |
| AINVA7    | U1 | EU1 | AINVA7   | 0 |
| NAIDINV1  | U1 | EU1 | NAIDINV  | 0 |
| NAIDINV2  | U1 | EU1 | NAIDINV  | 0 |
| NAIDINV3  | U1 | EU1 | NAIDINV  | 0 |
| DESPROD1  | U1 | EU1 | DESPRODA | 0 |
| DESPROD2  | U1 | EU1 | DESPRODA | 1 |
| DESPROD3  | U1 | EU1 | DESPROD3 | 0 |
| DESPROD4  | U1 | EU1 | DESPRODB | 1 |
| DESPROD5  | U1 | EU1 | DESPRODB | 0 |
| DESPROD6  | U1 | EU1 | DESPROD6 | 0 |
| CARPH1    | U1 | EU1 | CARPHA   | 0 |
| CARPH2    | U1 | EU1 | CARPHA   | 0 |
| CARPH3    | U1 | EU1 | CARPH3   | 0 |
| CARPH4    | U1 | EU1 | CARPH4   | 0 |
| CARPH5    | U1 | EU1 | CARPHB   | 0 |
| CARPH6    | U1 | EU1 | CARPHB   | 0 |
| CARPH7    | U1 | EU1 | CARPHB   | 0 |
| CARPH8    | U1 | EU1 | CARPHB   | 0 |
| CHARPH1   | U1 | EU1 | CHARPHA  | 0 |
| CHARPH2   | U1 | EU1 | CHARPHA  | 0 |
| CHARPH3   | U1 | EU1 | CHARPHA  | 0 |
| CHARPH4   | U1 | EU1 | CHARPH4  | 0 |
| CHARPH5   | U1 | EU1 | CHARPHB  | 0 |
| CHARPH6   | U1 | EU1 | CHARPHB  | 0 |
| AMELPH1   | U1 | EU1 | AMELPH1  | 0 |
| AMELPH2   | U1 | EU1 | AMELPH2  | 0 |
| RENPH1    | U1 | EU1 | RENPH1   | 0 |
| RENPH2    | U1 | EU1 | RENPH    | 0 |
| RENPH3    | U1 | EU1 | RENPH    | 0 |
| ACTPROD   | U1 | EU1 | ACTPROD  | 1 |
| ACHCESS1  | U1 | EU1 | ACHCESS  | 0 |
| ACHCESS1  | U1 | EU1 | ACHCESS  | 0 |
| STOCKS1   | U1 | EU1 | STOCKS   | 0 |
| STOCKS2   | U1 | EU1 | STOCKS   | 0 |
| CONSOM1   | U1 | EU1 | CONSOM1  | 0 |



|          |    |     |          |   |
|----------|----|-----|----------|---|
| CONSOM2  | U1 | EU1 | CONSOM2  | 0 |
| CONSOM3  | U1 | EU1 | CONSOM   | 0 |
| CONSOM4  | U1 | EU1 | CONSOM   | 0 |
| PROD1    | U1 | EU1 | PRODUA   | 0 |
| PROD2    | U1 | EU1 | PRODUA   | 1 |
| PROD3    | U1 | EU1 | PRODUA   | 0 |
| PROD4    | U1 | EU1 | PRODUB   | 0 |
| PROD5    | U1 | EU1 | PRODUB   | 1 |
| VENTESV1 | U1 | EU1 | VENTESV  | 0 |
| VENTESV2 | U1 | EU1 | VENTESV  | 0 |
| VENTESV3 | U1 | EU1 | VENTESV  | 0 |
| POLL1    | U1 | EU1 | POLL1    | 0 |
| POLL2    | U1 | EU1 | POLL2    | 0 |
| POLL3    | U1 | EU1 | POLL3    | 0 |
| POLL4    | U1 | EU1 | POLL4    | 0 |
| STRPRG   | U1 | EU1 | STRPRG   | 0 |
| STRPRPRO | U1 | EU1 | STRPRPRO | 0 |
| PRODUCTI | U1 | EU1 | PRODUCTI | 1 |
| RESEX1   | U1 | EU1 | RESEXA   | 1 |
| RESEX2   | U1 | EU1 | RESEXA   | 0 |
| RESEX3   | U1 | EU1 | RESEXA   | 0 |
| RESEX4   | U1 | EU1 | RESEXB   | 0 |
| RESEX5   | U1 | EU1 | RESEXB   | 1 |
| RESEX6   | U1 | EU1 | RESEXB   | 0 |
| DISBEN1  | U1 | EU1 | DISBEN   | 0 |
| DISBEN2  | U1 | EU1 | DISBEN   | 1 |
| DISBEN3  | U1 | EU1 | DISBEN   | 1 |
| DISBEN4  | U1 | EU1 | DISBEN4  | 1 |
| INDP     | U1 | EU1 | INDP     | 0 |
| IMPOS    | U1 | EU1 | IMPOS    | 0 |

7) Relation UTILISATEUR (USERID,PASSWORD,NOM)

=====

|    |        |                |
|----|--------|----------------|
| U1 | NAMUR1 | SCHMIT ROGER   |
| U2 | NAMUR2 | DEVULDER ALAIN |
| U3 | NAMUR3 | IBM FRANCE     |
| U4 | NAMUR4 | FNDP NAMUR     |

8) Relation VERSIONS (RELPHID,PREF,ETAT,CARDIN)

=====

|          |      |   |   |
|----------|------|---|---|
| DONSS    | 77 1 | P | 1 |
| DPP      | 77 1 | P | 1 |
| DIS      | 77 1 | P | 1 |
| CAPASSIF | 76   | C | 5 |
| CAPASSIF | 77   | C | 5 |
| CAPP1    | 76   | C | 5 |
| CAPP1    | 77   | C | 5 |
| CAPP2    | 76   | P | 2 |
| CAPP3    | 70   | P | 1 |
| CAPP3    | 73   | P | 1 |
| CAPP3    | 72   | P | 1 |

|           |      |   |   |
|-----------|------|---|---|
| CAPP3     | 74   | P | 1 |
| CAPP4     | 77   | P | 2 |
| CAPP5     | 77   | P | 2 |
| CAPP6     | 77   | P | 2 |
| CAPP7     | 77   | P | 2 |
| CAPP8     | 77   | P | 2 |
| CAPE1     | 76   | C | 5 |
| CAPE1     | 77   | C | 5 |
| CAPE2     | 76   | P | 1 |
| CAPE2     | 77   | P | 1 |
| CAPACTIF  | 76   | C | 5 |
| CAPACTIF  | 77   | C | 5 |
| CAPANIMM1 | 76   | C | 5 |
| CAPANIMM1 | 77   | C | 5 |
| CAPANIMM2 | 77   | C | 5 |
| CAPAIMM1  | 76   | C | 5 |
| CAPAIMM1  | 77   | C | 5 |
| CAPAIMM2  | 77   | P | 2 |
| CAPHYS1   | 77   | C | 6 |
| CAPHYS2   | 77   | P | 7 |
| CAPHYS3   | 77   | P | 8 |
| CAPHYS4   | 77   | P | 6 |
| CAPHYS5   | 77   | P | 6 |
| RATFIN1   | 76   | C | 5 |
| RATFIN1   | 77   | C | 5 |
| RATFIN2   | 76   | C | 5 |
| RATFIN2   | 77   | C | 5 |
| RATFIN3   | 76   | C | 5 |
| RATFIN3   | 77   | C | 5 |
| RATFIN4   | 77   | C | 5 |
| CAPINV1   | 77   | C | 6 |
| CAPINV2   | 77   | P | 2 |
| AINVA1    | 76   | P | 1 |
| AINVA2    | 77   | P | 1 |
| AINVA3    | 77   | P | 2 |
| AINVA4    | 77   | P | 1 |
| AINVA5    | 77   | P | 1 |
| AINVA7    | 77   | P | 1 |
| DESPROD1  | 77   | P | 1 |
| DESPROD2  | 77   | C | 6 |
| DESPROD4  | 77   | C | 5 |
| DESPROD5  | 77   | C | 5 |
| CARPH1    | 77 4 | C | 6 |
| CARPH1    | 77 3 | P | 3 |
| CARPH1    | 77 2 | P | 3 |
| CARPH1    | 77 1 | P | 3 |
| CARPH2    | 77 1 | P | 4 |
| CARPH3    | 70 1 | P | 2 |
| CARPH4    | 77 2 | P | 5 |
| CHARPH1   | 77 4 | C | 6 |
| CHARPH2   | 77 2 | C | 6 |
| CHARPH3   | 77 2 | C | 6 |
| AMELPH1   | 77   | P | 1 |
| RENPH2    | 77   | P | 1 |
| RENPH3    | 77   | P | 1 |
| ACTPROD   | 77   | C | 6 |
| ACHCESS1  | 77   | C | 6 |
| ACHCESS2  | 77   | C | 6 |



|          |      |   |   |
|----------|------|---|---|
| STOCKS1  | 77   | C | 6 |
| STOCKS2  | 77   | C | 6 |
| CONSOM1  | 77   | P | 7 |
| CONSOM2  | 77   | P | 7 |
| CONSOM3  | 77   | C | 6 |
| CONSOM4  | 77   | C | 6 |
| PROD1    | 77   | C | 6 |
| PROD2    | 77   | C | 6 |
| PROD3    | 77   | P | 2 |
| PROD4    | 77   | P | 8 |
| PROD5    | 77   | P | 3 |
| VENTESV1 | 76   | P | 1 |
| VENTESV1 | 77   | P | 8 |
| VENTESV2 | 77   | P | 7 |
| VENTESV3 | 77   | C | 6 |
| STRPRG   | 77   | C | 6 |
| STRPRPRO | 77   | P | 7 |
| PRODUCTI | 77   | C | 6 |
| PRODUCTI | 76   | P | 1 |
| RESEX1   | 77   | C | 5 |
| RESEX2   | 77   | C | 5 |
| RESEX3   | 77   | C | 5 |
| RESEX4   | 77   | C | 6 |
| RESEX5   | 77   | C | 6 |
| DISBEN1  | 77   | C | 5 |
| DISBEN2  | 77   | C | 5 |
| DISBEN3  | 76   | P | 1 |
| DISBEN3  | 77   | C | 5 |
| DISBEN3  | 78   | C | 6 |
| INDP     | 7712 | C | 1 |
| INDP     | 7801 | C | 1 |
| IMPOS    | 77   | C | 6 |

ANNEXE 4 :

EXEMPLES DE MISE A JOUR DE  
=====

RELATIONS DE LA BASE DE DONNEES  
=====



EX 1 : mise à jour de la relation CAPP2 (6.2)

====

MAJ

entrez relation,writepass

CAPP2,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'CAPP2'

composantes de la relation 'CAPP2' :

EA SUBEQ ACTN ACTA APPACT RESPR

liste des composantes à mettre à jour ?

SUBEQ ACTN ACTA

composantes qui seront modifiées :

SUBEQ ACTN ACTA

ok ? yes/no (y/n)

Y

la composante utilisée pour la recherche sera 'EA'

EA signifie : ENTREPRISE (11), ANNEE (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : EA =

.....

0042345670276

CAPP2 ?

SUBEQ : 555

?

R/575

ACTN : 4

?

R/5

ACTA : 0

?

C/0/1/

fichier FCAPP2 DBER A1 mis à jour

relation CAPP2 mise à jour

égalité SUBEQ dans la formule PC8

fichier FRATFIN1 DBER A1 mis à jour

résultat du calcul PC8 : 3850.33112

pas de calcul à faire pour ACTN

pas de calcul à faire pour ACTA

argument de recherche : EA =

.....

\*

temps de la mise à jour :

0 H 4 M 20 S 33

encore des données à mettre à jour

yes/no (y/n) ?

N



EX 2 : mise à jour de la relation DONSS (3.1)

=====

MAJ

entrez relation,writepass

DONSS,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'DONSS'

composantes de la relation 'DONSS' :

EAT MOISRO MONTANTO

liste des composantes à mettre à jour ?

#

toutes les composantes seront modifiées

la composantes utilisée pour la recherche sera 'EAT'

EAT signifie : ENTREPRISE (11), ANNEE (2), TRIMESTRE (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : EAT =

.....

0041234560177 1

DONSS ?

EAT : 0041234560177 1

?

MOISRO : 1

?

C/1/2/

MONTANTO : 100000

?

C/10/20/

fichier FDETONSS DBER A1 mis à jour

relation DONSS mise à jour

pas de calcul à faire pour MOISRO

pas de calcul à faire pour MONTANTO

argument de recherche : EAT =

.....

\*

temps de la mise à jour :

0 H 1 M 39 S 981

encore des données à mettre à jour

yes/no (y/n) ?

N

EX 3 : mise à jour de la relation CAPE1 (11.1)  
 ====

MAJ

entrez relation,writepass

CAPE1,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'CAPE1'

composantes de la relation 'CAPE1'

EA VMLT VCT

liste des composantes à mettre à jour ?

VMLT VCT

composantes qui seront modifiées :

VMLT VCT

ok ? yes/no (y/n)

y

la composante utilisée pour la recherche sera 'EA'

EA signifie : ENTREPRISE (11), ANNEE (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : EA =

.....

0041234560177

CAPE1 ?

VMLT : 2000000

?

C/20/25/

VCT : 6000000

?

C/60/65/

fichier FCAPE1 DBER A1 mis à jour

relation CAPE1 mise à jour

égalité VMLT dans la formule PC10

fichier FRATFIN1 DBER A1 mis à jour

résultat du calcul PC10 : 19.4080543

égalité VCT dans la formule PC9

fichier FRATFIN1 DBER A1 mis à jour

résultat du calcul PC9 : 1.18333333



argument de recherche : EA =

.....

\*

temps de la mise à jour :

0 H 6 M 14 S 861

encore des données à mettre à jour

yes/no (y/n) ?

y

entrez relation,writepass

...

... etc ...

...

EX 4 : mise à jour de la relation DESPROD1 (31.1)

====

MAJ

entrez relation,writepass

DESPROD1,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'DESPROD1'

composantes de la relation 'DESPROD1' :

BA CONSCAP

liste des composantes à mettre à jour ?

CONSCAP

composantes qui seront modifiées :

CONSCAP

ok ? yes/no (y/n)

Y

la composante utilisée pour la recherche sera 'BA'

BA signifie : ETABLISSEMENT (11), ANNEE (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : BA =

.....

0041234560177

DESPROD1 ?

CONSCAP : 40000

?

C/4/5/

fichier FDESPR01 DBER A1 mis à jour

relation DESPROD1 mise à jour

égalité CONSCAP dans la formule PC25

fichier FDESPR02 DBER A1 mis à jour

résultat du calcul PC25 : 0.0034722

argument de recherche : BA =

.....

\*

temps de la mise à jour :

0 H 3 M 8 S 764

encore des données à mettre à jour

yes/no (y/n)

N



EX 5 : mise à jour de la relation RENPH1 (44)

====

MAJ

entrez relation,writepass

RENPH1,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'RENPH1'

composantes de la relation 'RENPH1'

EOAMJ DVAC DES SOURCE

liste des composantes à mettre à jour ?

DES SOURCE

composantes qui seront modifiées :

DES SOURCE

ok ? yes/no (y/n)

Y

la composante utilisée pour la recherche sera 'EOAMJ'

EOAMJ signifie : ETABLISSEMENT (11), EMPLOI OFFERT (15), ANNEE (2),

MOIS (2), JOUR (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : EOAMJ =

.....

02456789005COMPTABLE 771201

RENPH1 ?

DES : COMPATBILITE

?

C/PAT/PTA/

SOURCE : JOURNAL

?

C/AL/AUX/

fichier FRENPH1 DBER A1 mis à jour

relation RENPH1 mise à jour

pas de calcul à faire pour DES

pas de calcul à faire pour SOURCE

argument de recherche : EOAMJ =

.....

\*  
temps de la mise à jour :

0 H 2 M 54 S 326

encore des données à mettre à jour  
yes/no (y/n)

N

EX 6 : mise à jour de la relation CARPH1 (35.1)

=====

MAJ

entrez relation,writepass

CARPH1,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'CARPH1'

composantes de la relation 'CARPH1' :

BAT NBTRAV NBMAN NBINT NBHOM NBFEM NBMVU NBEVUC NVPC

liste des composantes à mettre à jour ?

NBTRAV NBFEM

composantes qui seront modifiées :

NBTRAV NBFEM

ok ? yes/no (y/n)

y

la composante utilisée pour la recherche sera 'BAT'

BAT signifie : ETABLISSEMENT (11), ANNEE (2), TRIMESTRE (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : BAT =

.....

0041234560177 4

CARPH1 ?

NBTRAV : 5

?

C/5/6/

NBFEM : 3

?

R/4

fichier FCAPOHU1 DBER A1 mis à jour

relation RENPH1 mise à jour

égalité NBTRAV dans la fomule PC30

fichier FPRODUCT DBER A1 mis à jour

résultat du calcul PC30 : 180000

égalité NBTRAV dans la formule PC33

fichier FPRODUCT DBER A1 mis à jour

résultat du calcul PC33 : 2880000



égalité NBTRAV dans la formule PC17  
fichier FRATFIN3 DBER A1 mis à jour  
résultat du calcul PC17 : 0.225

égalité NBTRAV dans la formule PC26  
fichier FDESPR04 DBER A1 mis à jour  
résultat du calcul PC26 : 1600000

égalité NBTRAV dans la formule PC27  
fichier FDESPR04 DBER A1 mis à jour  
résultat du calcul PC27 : 1500000

égalité NBTRAV dans la formule PC51  
fichier FPRODUCT DBER A1 mis à jour  
résultat du calcul PC51 : 213,8

égalité NBTRAV dans la formule PC52  
fichier FPRODUCT DBER A1 mis à jour  
résultat du calcul PC52 : 1.8

mise à jour lors des calculs d' une donnée primaire : DEPTR

égalité DEPTR dans la formule PC31  
fichier FPRODUCT DBER A1 mis à jour  
résultat du calcul PC31 : 0.00001

mise à jour lors des calculs d' une donnée primaire : VAMT

égalité VAMT dans la formule PC34  
fichier FPRODUCT DBER A1 mis à jour  
résultat du calcul PC34 : 37.402597

mise à jour lors des calculs d' une donnée primaire : PMVO

égalité PMVO dans la formule PC53  
fichier FPRODUC2 DBER A1 mis à jour  
résultat du calcul PC53 : 0.00641406

mise à jour lors des calculs d' une donnée primaire : PMVA

égalité PMVA dans la formule PC31  
fichier FPRODUCT DBER A1 mis à jour  
résultat du calcul PC31 : 0.00001

mise à jour lors des calculs d' une donnée primaire : PRM

égalité PRM dans la formule PC32  
fichier FPRODUCT DBER A1 mis à jour  
résultat du calcul PC32 : 0.00011111

argument de recherche BAT =

.....

\*

temps de la mise à jour

0 H 32 M 30 S 923

encore des données à mettre à jour

yes/no (y/n)

N



EX 7 : mise à jour de la relation CONSOM3 (51.1) et  
 ==== de la relation CONSOM2 (50)

MAJ

entrez relation,writepass

CONSOM3,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'CONSOM3'

composantes de la relation 'CONSOM3' :

BA VGAC VCE VCER VCME VCPO QGAC QCE VFT VTST VCHFD VDER VPAV VLBFP  
 VLBEQ VFB

liste des composantes à mettre à jour ?

VCE QCE

composantes qui seront modifiées :

VCE QCE

ok ? yes/no (y/n)

y

la composante utilisée pour la recherche sera 'BA'

BA signifie : ETABLISSEMENT (11), ANNEE (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : BA =

.....

0043456780377

CONSOM3 ?

VCE : 9000

?

C/9/10/

QCE : 180

?

C/18/20/

fichier FCONSOM3 DBER A1 mis à jour

relation CONSOM3 mise à jour

pas de calcul à faire pour VCE

pas de calcul à faire pour QCE

argument de recherche : BA =

.....

\*

temps de la mise à jour :

0 H 6 M 0 S 780

encore des données à mettre à jour

yes/no (y/n) ?

Y

entrez relation,writepass

CONSOM2,EU1

relation ok \*\*\*

writepass ok \*\*\*

la relation à mettre à jour est 'CONSOM2'

composantes de la relation 'CONSOM2' :

ESEA QCONS VCONS

liste des composantes à mettre à jour ?

VCONS

composantes qui seront modifiées :

VCONS

ok ? yes/no (y/n)

Y

la composante utilisée pour la recherche sera 'ESEA'

ESEA signifie : ETABLISSEMENT (11), SOURCE D' ENERGIE (5), ANNEE (2)

pour terminer la mise à jour entrez '\*'

argument de recherche : ESEA =

.....

004456789044444477

pas trouvé

entrez '.CRE' pour création, retour de chariot pour continuer

argument de recherche : ESEA =

.....

\*

temps de la mise à jour :

0 H 8 M 8 S 654

encore des données à mettre à jour

yes/no (y/n) ?

NO



BUMP



0 0 3 2 1 2 9 2 7

\*FM B16/1979/14

